

PDP-11 FORTRAN-77

Installation Guide/Release Notes

Order No. AA-V196A-TK

August 1983

This guide describes the procedures for installing PDP-11 FORTRAN-77 on the RSX-11M/M-PLUS, RSTS/E, and VAX/VMS operating systems.

SUPERSESSION/UPDATE INFORMATION: This is a new manual for this release.

OPERATING SYSTEM AND VERSION: RSX-11M V4.1
RSX-11M-PLUS V2.1
RSTS/E V8.0
VAX/VMS V3.2

SOFTWARE VERSION: FORTRAN-77 V5.0

digital equipment corporation • maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1983 by Digital Equipment Corporation
All Rights Reserved.

Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	RSX
DEC/CMS	EduSystem	UNIBUS
DEC/MMS	IAS	VAX
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	digital
DECUS	RSTS	
DECwriter		

ZK2391

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726 (Ottawa-Hull)
800-267-6146 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

*Any prepaid order from Puerto Rico must be placed
with the local Digital subsidiary (809-754-7575)

DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: A&SG Business Manager

DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation
A&SG Business Manager
c/o Digital's local subsidiary or
approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

CONTENTS

	Page
PREFACE	vii
 CHAPTER 1	 INTRODUCTION
1.1	SYSTEM REQUIREMENTS 1-1
1.1.1	RSX-11M 1-1
1.1.2	RSX-11M-PLUS 1-2
1.1.3	RSTS/E 1-2
1.1.4	VAX/VMS 1-2
1.2	DISTRIBUTION FILES 1-3
1.2.1	[1,2] 1-3
1.2.2	[11,36] 1-3
1.2.3	[11,37] 1-3
1.2.4	[11,40] 1-4
 CHAPTER 2	 INSTALLATION PLANNING
2.1	SELECTING THE DEFAULT FORTRAN 2-1
2.1.1	Selecting F77 As the Default FORTRAN 2-1
2.1.2	Selecting FOR as the Default FORTRAN 2-2
2.2	SELECTING F77 COMPILER OPTIONS 2-2
2.3	SELECTING F77 OTS OPTIONS 2-2
2.4	PRELIMINARY INSTALLATION PROCEDURES 2-3
2.4.1	RSX-11M/M-PLUS 2-4
2.4.2	RSTS/E 2-5
2.4.3	VAX/VMS 2-5
2.5	MAKING THE COMPILER TASK AVAILABLE 2-5
2.5.1	RSX-11M/M-PLUS 2-5
2.5.2	RSTS/E 2-5
 CHAPTER 3	 SYSTEM TAILORING
3.1	OPTIONS AFFECTING COMPILE-TIME PERFORMANCE 3-1
3.1.1	Additional Temporary Files 3-1
3.1.2	Selecting the Size of the Dynamic Storage Area 3-2
3.1.2.1	Operating Systems Supporting Dynamic Memory Allocation 3-2
3.1.2.2	RSX-11M Without Dynamic Memory Allocation 3-4
3.2	OTS OPTIONS 3-4
3.2.1	F4P11S 3-4
3.2.2	Short Error Text - RSX-11M/M-PLUS and RSTS/E Only 3-5
3.2.3	F4PMAP 3-6
3.2.4	F4PEIS 3-7
3.2.5	F4PCVF 3-7
3.2.6	F4PNER 3-7
3.2.7	F4PNIO 3-8
3.2.8	F4PRAN 3-8
3.2.9	F4PRES 3-8
3.2.10	OTS Overlay Description Files 3-8
 CHAPTER 4	 RSX-11M/M-PLUS INSTALLATION
4.1	INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION 4-1
4.1.1	Preparations 4-1

CONTENTS

4.1.2	Building the Compiler	4-2
4.1.3	Building the OTS	4-2
4.1.4	Copying the Message File	4-3
4.2	INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION	4-4
4.2.1	Preparations	4-4
4.2.2	Building the Compiler	4-4
4.2.3	Building the OTS	4-5
4.2.4	Copying the Message File	4-6
4.3	INVOKING THE COMPILER	4-6
CHAPTER 5	RSTS/E INSTALLATION	
5.1	MOUNTING (AND COPYING) THE AUTOPATCH KIT	5-1
5.1.1	Mounting and Copying a Magnetic-Tape Kit	5-1
5.1.2	Mounting (and Copying) a Disk Kit	5-2
5.2	MOUNTING THE F77 DISTRIBUTION KIT	5-2
5.3	BUILDING THE COMPILER AND THE OTS	5-2
CHAPTER 6	INSTALLATION ON VAX/VMS UNDER AME	
6.1	PREPARATION	6-1
6.2	BUILDING THE COMPILER	6-1
6.3	BUILDING THE OTS	6-3
6.4	INSTALLATION VERIFICATION	6-3
6.5	CLEAN-UP	6-4
CHAPTER 7	SYSTEM BUILD VERIFICATION	
7.1	TEST PROGRAM	7-1
7.2	TEST RESULTS	7-2
7.3	VERIFICATION PROCEDURE FAILURE	7-2
CHAPTER 8	OTS RESIDENT LIBRARIES	
8.1	TYPES OF RESIDENT LIBRARIES	8-1
8.1.1	Noncluster Libraries	8-1
8.1.2	Cluster Libraries	8-2
8.2	CREATING AN OTS RESIDENT LIBRARY	8-2
8.2.1	General Procedure	8-3
8.2.2	Building a Noncluster Library with FCS Routines Included	8-4
8.2.3	Building a Noncluster Library Linked to FCSFSL	8-5
8.2.4	Building a Library to Cluster with RMSRES	8-6
8.2.5	Building a Library to Cluster with FCSRES	8-7
CHAPTER 9	RELEASE NOTES	
9.1	FORTRAN-77 OTS WITH RMS-11 SUPPORT	9-1
9.2	INSTALLING F77 ON RSTS/E	9-1
9.3	F77 TYPING OF RESULT-GENERIC FUNCTIONS	9-1
9.4	COMPILER FAILS TO GENERATE WARNING DIAGNOSTIC	9-2
9.5	ADJUSTABLE ARRAY USED IN INVALID CONTEXT	9-2
9.6	INTEGER ARRAY REFERENCE USED AS INDEX IN BYTE ARRAY SUBSCRIPT	9-2
9.7	COMMON SUBEXPRESSIONS WITHIN CONDITIONAL STATEMENT	9-3
9.8	RSTS/E INSTALLATION	9-3
APPENDIX A	ALTERNATIVE RSTS/E INSTALLATION	
A.1	INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION	A-1

CONTENTS

A.1.1	Preparations	A-1
A.1.2	Building the Compiler	A-2
A.1.3	Building the OTS	A-2
A.1.4	Copying the Message File	A-3
A.2	INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION . .	A-3
A.2.1	Preparations	A-4
A.2.2	Building the Compiler	A-4
A.2.3	Building the OTS	A-5
A.2.4	Copying the Message File	A-6
A.3	INVOKING THE COMPILER	A-6

APPENDIX B COMPILER TASK-BUILD FILES

B.1	PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR RSX-11M/M-PLUS (F7711M.CMD)	B-1
B.2	PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR RSTS/E (F77RST.CMD)	B-4
B.3	PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR VAX/VMS UNDER AME	B-7

INDEX

FIGURES

3-1	Compiler Performance	3-3
-----	--------------------------------	-----

PREFACE

MANUAL OBJECTIVES

This manual describes the procedures used to install PDP-11 FORTRAN-77 on any PDP-11 system under an RSX-11M, RSX-11M-PLUS, or RSTS/E operating system, from magnetic tape or disk cartridge distribution kits. PDP-11 FORTRAN-77 can operate on any PDP-11 processor with a Floating-Point Processor (FP11) or floating-point microcode option.

INTENDED AUDIENCE

This manual is intended for the PDP-11 system manager. You should be thoroughly familiar with the host operating system before installing the PDP-11 FORTRAN-77 software.

STRUCTURE OF THIS DOCUMENT

This manual is organized as follows:

- Chapter 1 presents the minimum system requirements for PDP-11 FORTRAN-77 installation on each of the host operating systems and lists the software files supplied in the distribution kits.
- Chapter 2 presents various system options and planning alternatives that must be considered before starting the actual installation process.
- Chapter 3 explains the effect of selecting certain compiler options described in Chapter 2 and details the optional Object Time System (OTS) modules distributed on the release media.
- Chapter 4 describes the installation procedures relevant to the RSX-11M and RSX-11M-PLUS operating systems.
- Chapter 5 describes the installation procedures on a RSTS/E operating system.
- Chapter 6 describes the installation procedures on a VAX/VMS system under the AME.
- Chapter 7 explains the testing procedure used to verify that your system is correctly installed.
- Chapter 8 describes OTS resident libraries.
- Chapter 9 contains release notes.
- Appendix A presents an alternative installation procedure for RSTS/E operating systems.
- Appendix B presents listings of the compiler task-build files for each system.

ASSOCIATED DOCUMENTS

The PDP-11 FORTRAN-77 User's Guide provides detailed information on using the FORTRAN-77 system. The PDP-11 FORTRAN-77 Language Reference Manual describes the elements of the FORTRAN language as implemented in FORTRAN-77. For information on the Object Time System, consult the PDP-11 FORTRAN-77 Object Time System Reference Manual. For information on how to use RSX-11M on VAX/VMS, consult the VAX-11/RSX-11M User's Guide. For a complete list of software documents, see your host operating-system documentation directory.

CONVENTIONS USED IN THIS MANUAL

The following documentation conventions are used throughout this manual:

- | | |
|------------------------|---|
| \$ ALTMODE | The symbol \$ represents the nonprinting ALTMODE key. This key is called the escape (ESC) key on many terminals. This key is pressed in place of the RETURN key when specified. Unless otherwise specified (that is, with the \$), all commands terminate with a carriage return. |
| CTRL/X | The notation CTRL/X indicates that you must press the CTRL (control) key while simultaneously typing a letter key (for example, CTRL/C, CTRL/Y, CTRL/O). |
| UPPERCASE
lowercase | In examples, text printed in UPPERCASE characters indicates literal information that must be entered as shown. Text printed in lowercase characters indicates that you are to substitute a word or value of your choice (for example, MTn:). |
| ␣ | Represents a carriage return. |

CHAPTER 1

INTRODUCTION

Section 1.1 of this chapter lists, by operating system, the PDP-11 features and components necessary for successfully building a PDP-11 FORTRAN-77 system. Section 1.2 lists the files contained in the PDP-11 FORTRAN-77 distribution kits.

1.1 SYSTEM REQUIREMENTS

The following sections list, by operating system, the minimum PDP-11 system requirements for installing PDP-11 FORTRAN-77. You need read only the section relevant to your host operating system.

NOTE

The term floating-point processor (lowercase) is used in this manual as a generic term to encompass any device or system for processing floating-point data.

1.1.1 RSX-11M

The software included in the PDP-11 FORTRAN-77 distribution media requires the following PDP-11 system components for normal use on an RSX-11M operating system:

- A PDP-11 processor capable of running RSX-11M and equipped with a floating-point processor
- A minimal 22K-word partition for compilation
- A minimum of 366 blocks of contiguous on-line disk storage for the compiler task
- A minimum of 150 to 250 blocks of on-line disk storage for the Object Time System library
- A KT-11 memory management unit, if you use virtual arrays

Consult Chapter 4 for RSX-11M installation instructions.

1.1.2 RSX-11M-PLUS

The software included in the PDP-11 FORTRAN-77 distribution media requires the following PDP-11 system components for normal use on an RSX-11M-PLUS operating system:

- A PDP-11 processor capable of running RSX-11M-PLUS and equipped with a floating-point processor
- A minimal 22-K word partition for compilation
- A minimum of 366 blocks of contiguous on-line disk storage for the compiler task
- A minimum of 150 to 250 blocks of on-line disk storage for the Object Time System library

Consult Chapter 4 for RSX-11M-PLUS installation instructions.

1.1.3 RSTS/E

The software included in the PDP-11 FORTRAN-77 distribution media requires the following PDP-11 system components for normal use on an RSTS/E operating system:

- A PDP-11 processor capable of running RSTS/E and equipped with a floating-point processor
- A minimum of 22K words of user space (swap maximum) for installation and compilation
- RT-11 and RSX run-time system support
- A minimum of 366 blocks of contiguous on-line disk storage for the compiler task
- A minimum of 150 to 250 blocks of on-line disk storage for the Object Time System library

Consult Chapter 5 for RSTS/E installation instructions.

1.1.4 VAX/VMS

The software included in the PDP-11 FORTRAN-77 distribution media requires the following VAX/VMS system components for normal use on a VAX/VMS operating system:

- A VAX-11 processor capable of running the RSX-11M AME
- A minimum of 390 on-line disk blocks for the compiler task
- A minimum of 150 to 250 contiguous blocks of on-line disk storage for the object time library

See Chapter 6 for VAX/VMS installation instructions.

1.2 DISTRIBUTION FILES

The software necessary for building a PDP-11 FORTRAN-77 system for an RSX-11M, RSX-11M-PLUS, or RSTS/E operating system is distributed on magnetic tape and disk cartridge media. These media contain the files described in the following sections. The files are listed under the user file directory (UFD) on which they reside in the kit or under their project programmer number (PPN).

1.2.1 [1,2]

File Name	Description
F77COM.MSG	Compiler diagnostic messages file
F77.CTL	RSTS/E build command file

1.2.2 [11,36]

File Name	Description
F77.OLB	PDP-11 FORTRAN-77 compiler object module library
F7711M.CMD F7711M.ODL	PDP-11 FORTRAN-77 compiler-build command files for RSX-11M/M-PLUS
F77RST.CMD F77RST.ODL	PDP-11 FORTRAN-77 compiler-build command files for RSTS/E

1.2.3 [11,37]

File Name	Description
F4POTS.OBJ	Object Time System concatenated object modules
FCS11M.OBJ	Specific OTS modules for RSX-11M/M-PLUS and RSTS/E using FCS-11
RMS11M.OBJ	Specific OTS modules for RSX-11M/M-PLUS and RSTS/E using RMS-11
SHORT.OBJ	OTS short-error text module for RSX-11M/M-PLUS and RSTS/E

INTRODUCTION

1.2.4 [11,40]

File Name	Description
F4P11S.OBJ	Specific OTS modules for RSX-11S subset
F4PMAP.OBJ	PDF name-mapping concatenated object module
F4PCVF.OBJ	Optional OTS floating-point formatted output conversion routine
F4PEIS.OBJ	EIS replacements for OTS integer functions that make use of the FPP
F4PNIO.OBJ	Optional OTS modules without FORTRAN I/O capability
F4PNER.OBJ	Optional OTS error-reporting module
F4PRAN.OBJ	Optional random-number generator

OTS Assembly Parameter Files

F77.MAC
FPP.MAC
FPPDP.MAC
FCS.MAC
RMS.MAC
RSXM.MAC
RSXS.MAC

File Name	Description
F77TST.FTN	System installation verification test program
F4PRES.MAC	OTS shared-library source file
FCS11M.ODL	OTS overlay description file for RSX-11M/M-PLUS and RSTS/E using FCS-11
RMS11M.ODL	OTS overlay description file for RSX-11M/M-PLUS and RSTS/E using RMS-11

CHAPTER 2

INSTALLATION PLANNING

This chapter discusses some alternatives you must consider before you start the installation process.

2.1 SELECTING THE DEFAULT FORTRAN

If you wish to run both FORTRAN IV (FOR) and PDP-11 FORTRAN-77 (F77) on the same system, you must decide whether FOR or F77 is to be the default FORTRAN. You must make this decision because:

- When you build a task, object modules produced by the FOR compiler or from the FOR Object Time System (OTS) must not be combined with object modules produced by the F77 compiler or from the F77 OTS.
- The F77 OTS and the FOR OTS cannot be in the same object module library. (The F77 OTS module is named F4POTS.)

Normally, the default FORTRAN OTS is part of system object module library LB:[1,1]SYSLIB.OLB. (For RSTS/E, the system object module library is LB:SYSLIB.OLB.) The Task Builder searches this library automatically when linking a task. Either the FOR OTS or the F77 OTS can be located in this library.

If you choose to use both FORTRAN systems, you must build a separate library to contain whichever FORTRAN OTS you do not put in LB:[1,1]SYSLIB.OLB. To use this separate library, you must explicitly name it in a Task Builder command line. (The Task Builder always searches the library named in the invoking command line before searching SYSLIB.)

Before selecting a default FORTRAN, consider which of the two FORTRANs you would like to be easier to use. The FORTRAN whose OTS is in SYSLIB will not require an explicit OTS library reference at task-build time, and therefore will consistently save you time and effort.

2.1.1 Selecting F77 As the Default FORTRAN

If you select F77 as the default FORTRAN, you can continue to use the FOR OTS in one of two ways: by building a separate library containing only FOR OTS modules; or by renaming the current SYSLIB, if it contains FOR OTS modules, to LB:[1,1]FOROTS.OLB. Other routines not pertaining to FORTRAN IV may also be in SYSLIB. These routines would have to be referenced as modules in FOROTS.OLB if SYSLIB were renamed.

2.1.2 Selecting FOR as the Default FORTRAN

If you select FOR as the default FORTRAN, follow the instructions in the IAS/RSX FORTRAN IV Installation Guide for including the FOR OTS in SYSLIB. Then build a separate F77 OTS library, as described in "Building the OTS" in the installation chapter pertaining to your operating system (Chapters 4, 5, or 6).

2.2 SELECTING F77 COMPILER OPTIONS

The following options are available when you build the F77 compiler:

- You can assign the compiler work file to a device other than the system device, to enhance compiler performance.
- You can specify the number of pages that are to make up the dynamic storage area to be used by the compiler. A larger paging memory decreases work-file paging activity and enhances compiler performance.
- You can specify the number of lines per page and the line width for listing devices that will be using non-U.S. standard paper stock.
- You can specify compiler command switch default settings that differ from the DIGITAL-provided defaults.
- You can alter the values of the following compiler limits:
 - Number of actual arguments per CALL or function reference
 - Number of named COMMON blocks
 - Number of OPEN statement keywords
 - DO and block IF statement nesting depth
- You can choose the default setting for the /F77 compiler switch to be either /F77 or /NOF77.

You can edit the compiler-build file to select any of the above options. Documentation within the file describes the options available and any limitations on choices (see Appendix A). Chapter 3 describes the magnitude of the performance improvements that result from various combinations of the first two options.

2.3 SELECTING F77 OTS OPTIONS

The F77 Object Time System supports either of the following file systems for I/O processing:

- File Control Services (FCS-11), which supports sequential and direct access to sequential files
- Record Management Services (RMS-11), which supports sequential, direct, and keyed access to sequential, relative, and indexed files

When you build the OTS, you must choose either the FCS-11 file system or the RMS-11 file system; the F77 OTS does not support both file systems at one time.

INSTALLATION PLANNING

The distribution kit contains alternate files for incorporating either FCS-11 or RMS-11.

The files for FCS-11 support are FCS11M.OBJ.

The files for RMS-11 support are RMS11M.OBJ.

Sections 4.1.3 and 4.2.3 include a choice of commands that determines whether your system supports RMS-11 or FCS-11.

Section 3.2 describes OTS options that you can add after you install the basic system.

2.4 PRELIMINARY INSTALLATION PROCEDURES

The following utility programs are required for systems using MCR. If your system uses MCR, install these programs before you proceed with the rest of the installation:

```
PIP
FLX  (if using a magnetic tape distribution)
TKB
LBR
```

For systems using RSTS/E, the RT-11 and RSX run-time systems must be added. The following utility programs are required on a RSTS/E system:

```
PIP.SAV
TKB.TSK
LBR.TSK
```

The RT-11 run-time system is required for the installation process -- specifically to enable the use of PIP. The RT-11 run-time support is not required for using PDP-11 FORTRAN-77.

If you are replacing any version of FORTRAN IV-PLUS, the following preliminary steps are required:

1. If the FORTRAN IV-PLUS compiler is installed in the system and you are installing the F77 compiler as F4P, remove it with the following code:

```
REM ...F4P (for RSX-11M/M-PLUS)
RUN $UTILTY (FOR RSTS/E) #CCL F4P==
```

2. If you currently have FORTRAN IV-PLUS, the following files will be on the system disk:

[1,2] F4PCOM.MSG	(All operating systems)
[libuic] F4P.TSK	(RSX-11M/M-PLUS)
[1,2] F4P.TSK	(RSTS/E)

After PDP-11 FORTRAN-77 has been successfully installed, the following files will be on the system disks:

[1,2] F77COM.MSG	(All operating systems)
[libuic] F77.TSK	(RSX-11M/M-PLUS)
[1,2] F77.TSK	(RSTS/E)

INSTALLATION PLANNING

3. After PDP-11 FORTRAN-77 has been successfully installed and if you no longer intend to use FORTRAN IV-PLUS, you can delete the following files:

[1,2] F4PCOM.MSG	(All operating systems)
[libuic] F4P.TSK	(RSX-11M/M-PLUS)
[1,2] F4P.TSK	(RSTS/E)

4. Delete all files in reserved UFD [11,36], or, in the case of RSTS/E, the particular UFD on which FORTRAN IV-PLUS resides.
5. If you want to incorporate the F77 OTS into SYSLIB.OLB, obtain a copy of SYSLIB. Save the original SYSLIB.OLB by renaming it to SYSLIB.OLD. The renaming facilitates recovery of the original SYSLIB if the build is unsuccessful.

You must either obtain a copy of SYSLIB that contains everything currently in SYSLIB except the PDP-11 FORTRAN OTS and incorporate the PDP-11 FORTRAN-77 OTS into it, or obtain a fresh copy of SYSLIB from the operating system distribution kit and reincorporate all optional software modules. OTS modules cannot be added to a library containing a previous version of the OTS.

6. If you want to incorporate the F77 OTS into an existing library, make sure that the library module name table and the entry point name table can accommodate the OTS.

Module names: 200
Entry points: 450

If necessary, compress the existing library using the librarian compress command.

2.4.1 RSX-11M/M-PLUS

On an RSX-11M or RSX-11M-PLUS system, use a privileged user account -- for example, the account with UIC [1,1] -- for all operations used in building the FORTRAN-77 system.

During the build process, always run a program that is to create a file under the UIC that corresponds to the UFD in which the file will be created. You can use the SET command to set the UIC as required.

The F77 compiler is built into a system UIC called LIBUIC, which is reserved for DIGITAL products.

On RSX-11M systems, LIBUIC is equivalent to SYSUIC. You can find the value of SYSUIC by using the SET command as follows:

```
>SET /SYSUIC  
SYSUIC = [x,y]
```

On RSX-11M-PLUS systems, you can find the value of LIBUIC by using the SET command as follows:

```
>SET /LIBUIC  
LIBUIC = [x,y]
```

Use the value [x,y] as required during the installation.

INSTALLATION PLANNING

2.4.2 RSTS/E

On a RSTS/E system, use a privileged user account for all operations used in building the PDP-11 FORTRAN-77 system. Do not, however, use an account with a PPN of [0,0], [0,1], [1,1], or [1,2]. These accounts are reserved privileged accounts.

2.4.3 VAX/VMS

VAX/VMS under AME uses the system manager's account described in Chapter 6.

2.5 MAKING THE COMPILER TASK AVAILABLE

After the compiler task has been created, the task image file must be made available for use (installed). The following sections tell how to install the FORTRAN-77 compiler under each operating system.

2.5.1 RSX-11M/M-PLUS

On an RSX-11M or RSX-11M-PLUS system, you install the F77 compiler from the library area LIBUIC as follows:

```
>INS [libuic]F77
```

You can install the F77 compiler either in the saved system image or as part of system startup procedures. To install F77 during system startup, include the appropriate INS command (above) in system startup file [1,2]STARTUP.CMD. To install F77 in the saved system image, consult the RSX-11M System Generation and Management Guide or the RSX-11M-PLUS System Generation and Management Guide.

To install PDP-11 FORTRAN-77 as F4P, which may be desirable if your site has existing command files that call the F4P compiler, use the following command:

```
>INS [libuic] F77/TASK= ...F4P
```

2.5.2 RSTS/E

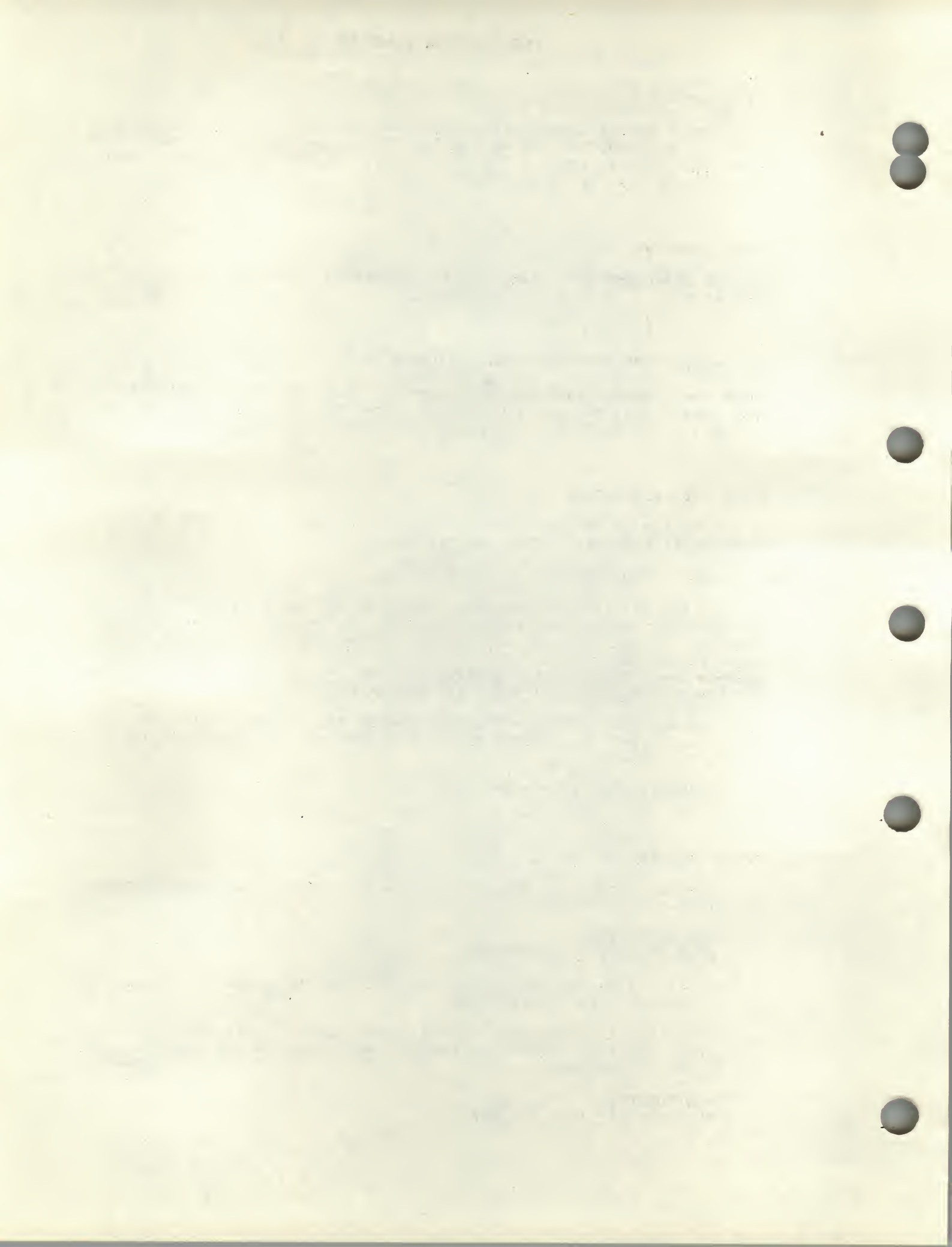
On a RSTS/E system, you may install the F77 compiler as a CCL command, as follows:

```
RUN $UTILTY  
#CCL F77==SY:[1,2]F77.TSK
```

To install F77 during system startup, include the above CCL command in system startup file [1,2]CCL.CMD.

To install PDP-11 FORTRAN-77 as F4P, which may be desirable if your site has existing command files that may call the F4P compiler, use the following commands:

```
RUN $UTILTY  
#CCL F4P==SY:[1,2]F77.TSK
```



CHAPTER 3

SYSTEM TAILORING

This chapter describes options you may choose when building PDP-11 FORTRAN-77 into your system. It includes factors affecting compiler performance and information about optional OTS modules that you can use to tailor PDP-11 FORTRAN-77 to your particular applications.

3.1 OPTIONS AFFECTING COMPILE-TIME PERFORMANCE

There are three options open to you that affect compile-time performance:

- You may choose one or two additional temporary disk files for the compiler to use for storing information during the compilation process.
- You may alter the size of the dynamic storage area in the compiler.
- You may place the compiler work file on a fixed-head disk.

The PDP-11 FORTRAN-77 compiler uses temporary disk files for storing information during the compilation process. The compiler requires at least one temporary file, called the work file.

The work file contains information that the compiler normally accesses at random (for example, the symbol table and the constants table). The dynamic storage area within the compiler is used to manipulate this information. (Only part of the work file is in memory at any given time. Software paging techniques move information back and forth between the dynamic storage area and the work file.)

Information must be moved into the dynamic storage area when needed by the compiler. Therefore, increasing the size of the dynamic storage area increases compilation speed by reducing the number of disk I/O operations (see Section 3.1.3). Similarly, speeding up the average disk I/O operation, by placing the work file on a fixed-head disk instead of a moving-head disk, also improves the compilation rate (see Section 3.1.6).

3.1.1 Additional Temporary Files

The /WF:w compiler switch specifies the number of temporary disk files that are to be available to the compiler. If you specify /WF:1, the compiler stores internal representations of optimized source code and PDP-11 code in just one file, the work file. However, if you specify /WF:2 (or /WF:3), the compiler stores some (or all) of these representations in the one or two other temporary files. (The /WF:2 option is the default.)

Using additional temporary files slows the compilation process, but it significantly increases the capacity of the compiler. For instance, with three temporary files (/WF:3) available to it, the compiler can compile a program that is approximately three times larger than any it can compile with only one temporary file (/WF:1).

No significant change occurs in the compilation rate if you place the temporary files on a fixed-head disk, because these files are written and read sequentially.

3.1.2 Selecting the Size of the Dynamic Storage Area

Increasing the size of the dynamic storage area increases the rate of compilation. Experimental data indicates that using 14 pages of dynamic storage results in optimal improvement in compilation speed.

Figure 3-1 illustrates the correlation between compile time and the size of the dynamic storage area. The compile time of four different FORTRAN programs, varying in length from 90 to 450 statements, was measured on a PDP-11/60. The compiler used two temporary files (/WF:2), with the work file residing on the system moving-head disk (RP04). The dynamic storage area varied in size from 4 to 26 pages.

No output listing was produced.

The measurements at the end points of each curve denote approximate compilation rate measured in statements compiled per minute. Continuation and comment lines were not counted.

As the figure shows, compilation speed is approximately three times greater when 26 pages of dynamic storage are used than it is when only 4 pages are used. However, using 14 pages results in optimal compiler performance. Building the compiler with more than 14 pages of dynamic storage achieves minimal improvement in the rate of compilation. The default size of the dynamic storage area is 12 pages.

3.1.2.1 Operating Systems Supporting Dynamic Memory Allocation - Under a RSTS/E, RSX-11M-PLUS, or RSX-11M system, with dynamic memory allocation, you specify the size of the PDP-11 FORTRAN-77 compiler's dynamic storage area by using the EXTTSK option in the task-build command file. The value specified by EXTTSK is the size of the dynamic storage area in decimal words. The size of the dynamic storage area is computed as follows:

$$256 * (n + w + 1)$$

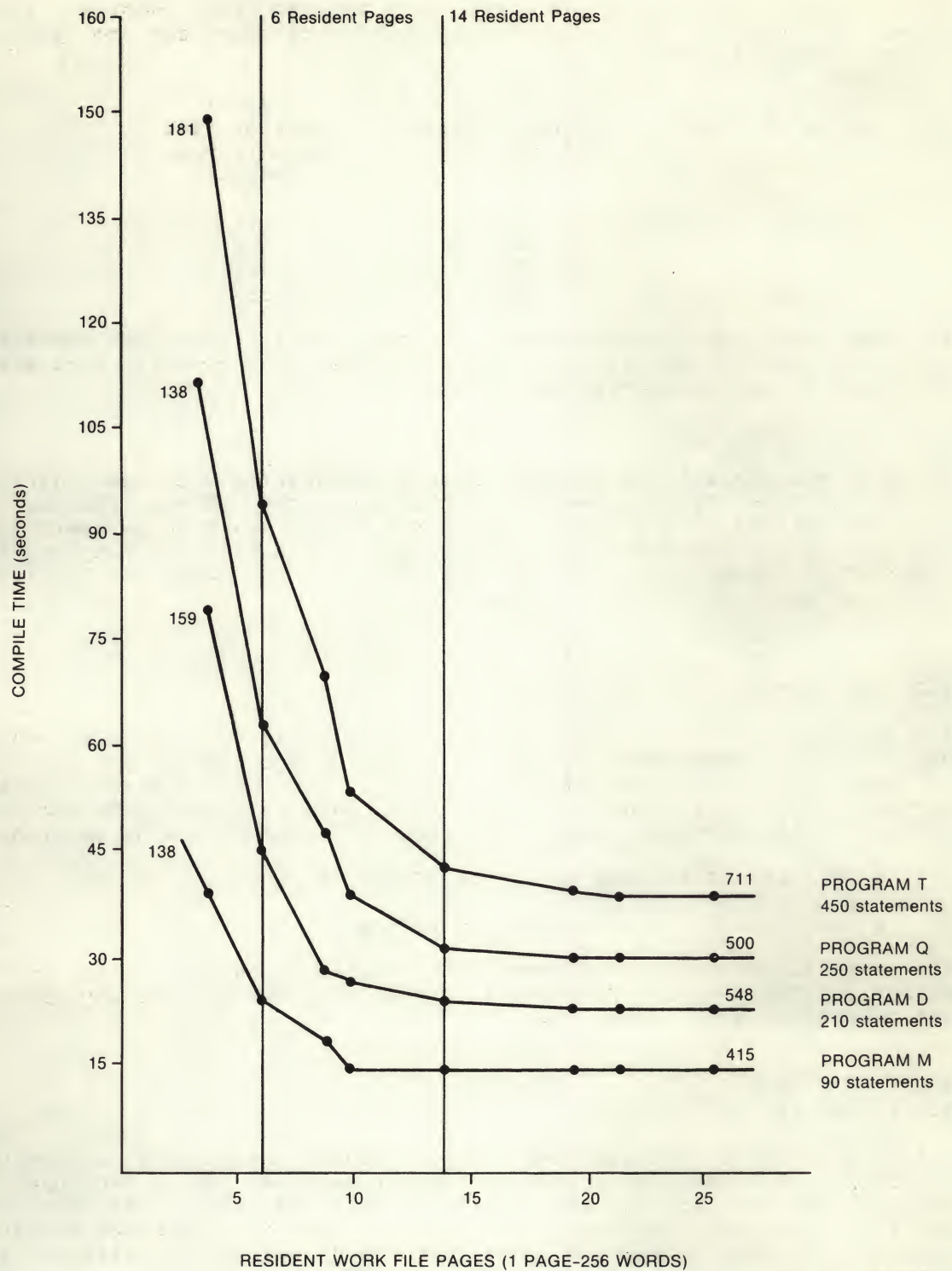
n

The number of pages for the dynamic storage area.

w

The value specified in the /WF:w switch.

You can override the dynamic storage area specified by EXTTSK at installation by means of the INC switch on the INSTALL (INS) command. The task extension size is specified in decimal words.



ZK-209-81

Figure 3-1: Compiler Performance

The following table shows the correlation between the compiler task size, the EXTTSK value, and the number of pages for the dynamic storage area under /WF:2.

Number of Pages	EXTTSK - Value INS/INC Value	Size of Compiler Task Compiler Task (Words)
4	1792	22K
8	2816	23K
12	3840	24K
16	4864	25K

For RSX-11M/M-PLUS installations that use by default the ANSI magnetic tape version of FCS-11 (LB:[1,1]ANSLIB.OLB), the compiler task size increases by approximately 500 words.

3.1.2.2 RSX-11M Without Dynamic Memory Allocation - On an RSX-11M system without dynamic memory allocation, the PDP-11 FORTRAN-77 compiler determines the size of the partition in which it is operating and uses all of the memory in that partition. Install the compiler in a partition large enough for the compiler to run with the desired number of pages of dynamic storage.

3.2 OTS OPTIONS

The distribution kit includes a number of optional OTS modules under UFD [11,40]. After building the OTS library, you can add one or more of these optional modules to the library, or you can maintain these modules separately and refer to them only as needed. To add the modules to the library, refer to procedures in the following sections:

- Sections 4.1.3 and 4.2.3 for RSX-11M/M-PLUS
- Sections A.1.3 and A.2.3 for RSTS/E

The installation procedures copy these modules to LB:[1,1] (LB: under RSTS/E). The PDP-11 FORTRAN-77 system does not require any of the optional modules for normal use.

3.2.1 F4P11S

F4P11S.OBJ is a concatenated object module containing alternate versions of FORTRAN sequential I/O support modules. These I/O support modules, designed for use with RSX-11S, provide sequential I/O to non-file-structured devices (for example, terminals, nonspooled card readers, and line printers). These modules do not use the file system but perform direct QIO operations; they reduce task size by approximately 2500 words.

NOTE

In the following sections, if you are using the RSTS/E system, replace all occurrences of LB:[1,1] with LB:.

You can use F4P11S.OBJ in two ways:

- You can include it as an object module at task-build time, as follows:

```
TKB>MAIN/FP=MAIN,LB:[1,1]F4P11S.OBJ
```

- Or you can build a separate F77 OTS library for RSX-11S use, LB:[1,1]F4P11S.OLB, in addition to the host operating system's OTS library. To do this, when building the OTS, substitute module LB:[1,1]F4P11S.OBJ (or, for RSTS/E, LB:F4P11S.OBJ) for the file system module selected in Section 2.3. For example, replace the reference to FCS11M.OBJ with LB:[1,1]F4P11S.OBJ.

Use this OTS library, rather than the host operating system's OTS library, when building tasks for RSX-11S, as follows:

```
TKB>MAIN/FP=MAIN,LB:[1,1]F4P11S/LB
```

3.2.2 Short Error Text - RSX-11M/M-PLUS and RSTS/E Only

For error messages, the FORTRAN-77 OTS references an error-text module containing ASCII text. If your operating system is RSX-11M, RSX-11M-PLUS, or RSTS/E, you can use a long or a short error-text module. The long error-text module requires approximately 1000 words of memory, whereas the alternate version (SHORT.OBJ) requires only one word of memory.

A task with the short error-text module built into it generates complete error reports, but omits the 1-line description of the error condition. The PDP-11 FORTRAN-77 User's Guide contains a complete list of OTS error numbers and message text.

The F77 OTS, as built in the following chapters, uses the long error-text module by default. You can build a task using the short error-text module by loading module \$SHORT from the library. Therefore, for example, if the F77 OTS is part of SYSLIB, you use the following command to build the short error-text module into a task, if your operating system is RSX-11M or RSX-11M-PLUS:

```
TKB>MAIN/FP=MAIN,LB:[1,1]SYSLIB/LB:$SHORT
```

If your operating system is RSTS/E, and F4POTS is resident in SYSLIB, you may use the short error-text module by typing the following command:

```
TKB MAIN/FP=MAIN,LB:SYSLIB/LB:$SHORT
```

You may prefer to use the short error-text as the system default. If so, reorder the steps in the OTS build procedures so that insertion of SHORT.OBJ into the library is the last step rather than the first; if you are using the BUILD procedure for a RSTS/E installation, simply answer the relevant interactive question by typing SHORT.

For example, instead of using the sequence of commands given in Sections 4.1.3 and A.1.3 for installing the F77 OTS into SYSLIB, order the commands as follows. [(If you are using a RSTS/E system, replace

all occurrences of [11,36] with the account where the object modules were copied during installation.))

```
LBR>LB:SYSLIB.OLB=[11,36]FCS11M.OBJ/RP
or
LBR>LB:SYSLIB.OLB=[11,36]RMS11M.OBJ/RP

LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=[11,36]F4POTS.OBJ/RP
LBR>LB:SYSLIB.OLB=[11,36]SHORT.OBJ/RP
```

Reorder the commands in Section 4.2.3 and A.2.3 as follows:

```
LBR>LB:SYSLIB.OLB=FP:[11,37]FCS11M.OBJ/RP
or
LBR>LB:SYSLIB.OLB=FP:[11,37]RMS11M.OBJ/RP

LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=FP:[11,37]F4POTS.OBJ/RP
LBR>LB:SYSLIB.OLB=FP:[11,37]SHORT.OBJ/RP
```

If the short error-text is the system default, you can use the complete error-text by explicitly loading module \$ERTXT from the library. The following command loads the \$ERTXT module for RSX-11M/M-PLUS:

```
TKB>MAIN/FP=MAIN,LB:[1,1]SYSLIB/LB:$ERTXT
```

For RSTS/E, the command to the Task Builder takes the following form:

```
TKB> MAIN/FP=MAIN,LB:SYSLIB/LB:$ERTXT
```

3.2.3 F4PMAP

Module F4PMAP.OBJ consists of a set of concatenated object modules that can be used to transform intrinsic function names into internal names at task-build time. (The PDP-11 FORTRAN-77 compiler transforms intrinsic function names into internal names at compile time.)

Without F4PMAP.OBJ, if a program written in MACRO-11 attempts to reference a PDP-11 FORTRAN-77 intrinsic function with the FORTRAN name of the function instead of the internal name, an unresolved reference will occur during task build.

For example, F4PMAP.OBJ maps the FORTRAN name SIN by means of the following module:

```
      .TITLE      $MSIN
SIN::  JMP        $SIN
      .END
```

F4PMAP.OBJ contains an object module similar to the above module for each of the PDP-11 FORTRAN-77 intrinsic functions.

You can build an F4PMAP library as follows:

```
(MCR command format)
  LBR>LB:[1,1]F4PMAP.OLB/CR:40.=LB:[1,1]F4PMAP.OBJ

(IAS PDS command format)
  PDS>LIBR CREATE/SIZE:40 LB:[1,1]F4PMAP LB:[1,1]F4PMAP

(RSTS/E command format)
  LBR>LB:F4PMAP.OLB/CR:40=LB:F4PMAP.OBJ
```


3.2.4 F4PEIS

F4PEIS.OBJ is a concatenated object module containing extended instruction set (EIS) versions of certain integer functions that normally use a floating-point processor. This module allows FORTRAN programs that do not do floating-point arithmetic to run on a machine that has the extended instruction set but not a floating-point processor. The modules provided in the F77 OTS use a floating-point processor for maximum efficiency in certain INTEGER*4 computations.

Use one of the following commands to replace at task-build time the normal modules in file INTEGER with their EIS versions:

```
(MCR command format)
TKB>INTEGER/-FP=INTEGER, LB:[1,1]F4PEIS.OBJ
```

```
(RSTS/E command format)
TKB>INTEGER/-FP=INTEGER, LB:F4PEIS.OBJ
```

3.2.5 F4PCVF

Object module F4PCVF.OBJ is an alternative module for performing formatted output of floating-point values under control of the D, E, F, and G field specifiers. The standard module provided as part of the F77 OTS uses multiple-precision, fixed-point integer techniques to maintain maximum accuracy during the conversion of data (FPP hardware is not used). The alternative module performs the same functions using the FPP hardware. It is approximately twice as fast as, but in some cases slightly less accurate than, the standard module.

You can substitute the F4PCVF module for the default conversion module as follows:

```
(MCR command format)
LBR>LB:[1,1]SYSLIB=LB:[1,1]F4PCVF/RP
```

```
(RSTS/E command format)
LBR>LB:SYSLIB=LB:F4PCVF/RP
```

3.2.6 F4PNER

Object module F4PNER.OBJ is an alternative module for reporting run-time errors. If you use this module, the error-message text report is suppressed. However, error processing and calls to ERRSET, ERRSNS, and ERRST continue to operate normally; only the logging of the message on the user's terminal is suppressed. The STOP and PAUSE statement messages are also suppressed. F4PNER.OBJ reduces task size by about 375 words over the standard module.

If you use F4PNER with F4P11S or F4PNIO -- other optional OTS modules -- a multiply defined symbol error may result during task-build. Two correct ways to use F4PNER with F4P11S or F4PNIO follow:

- Build F4P11S (or F4PNIO) and F4PNER as separate libraries and use them as follows:

```
TKB>MAIN/FP=MAIN, LB:[1,1]F4PNER/LB: $NERRL,
      LB:[1,1]F4P11S/LB, LB:[A,A]F4POTS/LB
```


SYSTEM TAILORING

- Build an OTS by incorporating F4P11S.OBJ instead of FCS11M.OBJ into F4POTS. Name the resulting library F4P11S.OLB, and build F4PNER as a separate library. Use those libraries as follows:

```
TKB>MAIN/FP=MAIN, LB:[1,1]F4PNER/LB: $NERRL,  
LB:[1,1]F4P11S/LB
```

3.2.7 F4PNIO

F4PNIO.OBJ is a concatenated object module containing alternative versions of certain OTS routines that are always present in the user task and that provide support for FORTRAN I/O operations. The alternate routines in F4PNIO.OBJ do not support FORTRAN I/O and reduce task size by approximately 1000 words for programs that do not require FORTRAN I/O (such as process control).

3.2.8 F4PRAN

F4PRAN.OBJ is a concatenated object module containing an alternative random-number generator that is compatible with previous releases of PDP-11 FORTRAN. If you require this random-number generator for compatibility purposes, include file LB:[1,1]F4PRAN.OBJ at task-build time.

3.2.9 F4PRES

F4PRES.MAC is a MACRO-11 source file containing global references to all modules of the OTS. An OTS resident library (shared global area, or SGA) is an option available to installations that have many FORTRAN-77 programs as well as PDP-11 FORTRAN-77 development. F4PRES.MAC contains the global references to the OTS and documentation on logical groups of OTS modules. This feature aids in the building of an OTS resident library that is tailored to the needs of a particular installation.

See Chapter 8 for more information about building and using OTS resident libraries.

3.2.10 OTS Overlay Description Files

The two OTS overlay description files are:

```
FCS11M.ODL - FCS-11 support for RSX-11M/M-PLUS and RSTS/E  
RMS11M.ODL - RMS-11(K) support for RSX-11M/M-PLUS, and RSTS/E
```

Each file is an ODL fragment file that you can use for overlaying the PDP-11 FORTRAN-77 OTS modules. Each file contains documentation that describes OTS options as well as procedures for using the file.

NOTE

If you are using the RSTS/E system, files FCS11M.ODL and RMS11M.ODL contain references to LB:[1,1]. All occurrences of LB:[1,1] should be changed to LB:.

CHAPTER 4

RSX-11M/M-PLUS INSTALLATION

This chapter describes the procedures for installing PDP-11 FORTRAN-77 on the RSX-11M and RSX-11M-PLUS operating systems. Read the sections relevant to the type of distribution kit you are using (Section 4.1 for the magnetic tape distribution and Section 4.2 for the disk cartridge distribution). Section 4.3 presents instructions for invoking the compiler.

The basic installation procedure for PDP-11 FORTRAN-77 consists of the following:

- Building the F77 compiler task from an object-module library
- Building an F77 OTS library from object modules
- Copying the compiler diagnostic-message file from the distribution medium to the system disk

4.1 INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION

The following sections explain how to build PDP-11 FORTRAN-77 from a magnetic tape distribution kit.

4.1.1 Preparations

UFD [11,36] is used in building the PDP-11 FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system disk with the following command:

```
>UFD SY:[11,36]
```

Once UFD [11,36] is present (or if already present), assign FP: as the logical name for the system device as follows:

```
>ASN SY:=FP:
```

Now place the distribution magnetic tape, write-locked, on drive 0, and load the magnetic tape handler, if not already resident, as follows (Note that the device name for some magnetic tape units is MM: or MS:. If you are using such a device, substitute MM: or MS: for MT: in the following commands):

```
>LOA MT:
```

RSX-11M/M-PLUS INSTALLATION

Now mount the magnetic tape. If you are using RSX-11M, allocate the magnetic tape as follows:

```
>ALL MT:
```

On RSX-11M-PLUS, the magnetic tape must be mounted as a foreign device, as follows:

```
>MOU MT:/FOR
```

4.1.2 Building the Compiler

You build the compiler from the object module library supplied on the distribution medium. First, set the UIC to [11,36] and copy the required files from the magnetic tape to the system disk, as follows:

```
>SET /UIC=[11,36]
>FLX SY:=MT0:F77.OLB
>FLX SY:=MT0:F7711M.*
```

Now edit compiler task-build command file [11,36]F7711M.CMD to select installation options, as described in Section 2.2 and Appendix B.

Then set the UIC to the system library area LIBUIC as follows:

```
>SET /UIC=[libuic]
```

The [libuic] is determined by the system you are on. If you are on an RSX-11M-PLUS system, you can get the correct UIC with the command:

```
>SET /LIBUIC
```

If you are on an RSX-11M system, the system UIC is [1,54].

Now build the compiler as follows:

```
>TKB @[11,36]F7711M
```

Retain the edited command files in [11,36] for reference.

4.1.3 Building the OTS

To build the OTS, first set the UIC to [11,36] and copy the required OTS files from the magnetic tape to the system disk, as follows:

```
>SET /UIC=[11,36]
>FLX SY:[11,36]=MT0:[11,37]*.OBJ
>SET /UIC=[1,1]
```


You can now build the F77 OTS in either of two ways:

1. By merging the OTS concatenated object modules into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, invoke and use the LBR utility program as follows:

```
LBR>LB:SYSLIB.OLB=[11,36]SHORT.OBJ/RP
LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=[11,36]F4POTS.OBJ/RP
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both (see Section 2.3) as follows:

```
LBR>LB:SYSLIB.OLB=[11,36]FCS11M.OBJ/RP
or
LBR>LB:SYSLIB.OLB=[11,36]RMS11M.OBJ/RP
```

2. By generating a separate library using the OTS concatenated object modules, as follows. (If you build a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

That is, set the UIC to that of the new library, for example >SET/UIC=[1,1], and invoke the LBR utility program as follows:

```
LBR>LB:F4POTS.OLB/CR:150.=[11,36]SHORT.OBJ
LBR>LB:F4POTS.OLB/DG:$ERTXT
LBR>LB:F4POTS.OLB=[11,36]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
LBR>LB:F4POTS.OLB=[11,36]FCS11M.OBJ
or
LBR>LB:F4POTS.OLB=[11,36]RMS11M.OBJ
```

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
>FLX LB:[1,1]=MT0:[11,40]*.OBJ,*.ODL,*.MAC
```

Then copy the verification test program to the system disk as follows:

```
>FLX LB:[1,1]=MT0:[11,40]*.FTN
```

4.1.4 Copying the Message File

At this point, set the UIC to [1,2] and copy the compiler message file to the system disk as follows:

```
>SET /UIC=[1,2]
>FLX LB:[1,2]=MT0:[1,2]F77COM.MSG/IM:64./BL:14.
```

4.2 INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION

The following sections explain how to build FORTRAN-77 from a disk cartridge distribution kit. If the system disk is not the same device type as that of the distribution disk, place the distribution disk in drive 0. If the system disk is the same device type as that of the distribution disk, use drive 0 for the system disk and drive 1 for the distribution disk.

In the directions that follow, logical-unit assignment Dxn:=FP: assigns the logical-device name FP: to the drive of the distribution disk. In this assignment, x can take the value K, M, or L, and n can take the value 0 or 1 (on the basis of the type of the system device). Use DK0 or DK1 for the RK05 distribution; use DM0 or DM1 for the RK06 or RK07 distribution; and use DL0 or DL1 for the RL01 or RL02 distribution.

4.2.1 Preparations

UFD [11,36] is used in building the PDP-11 FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system disk with the following command:

```
>UFD SY:[11,36]
```

Once UFD[11,36] is present (or if already present), load the disk handler, if not already resident, as follows:

```
>LOA Dx:                (not needed if the system device is the
                        same device type as that of the
                        distribution disk)
```

Now place the distribution disk cartridge in drive n (write-locked), and make the logical assignment and mount the volume as follows:

```
>ASN Dxn:=FP:
>MOU FP:F77
```

4.2.2 Building the Compiler

You build the compiler from the object module library on the distribution disk. First, set the UIC to [11,36] and copy the compiler-build files from the distribution disk to the system disk, as follows:

```
>SET /UIC=[11,36]
>PIP SY:=FP:F77.OLB
>PIP SY:=FP:F7711M.*
```

Now edit compiler task-build command file [11,36]F7711M.CMD to select installation options, as described in Section 2.2 and in Appendix B.

Then set the UIC to system-library area LIBUIC as follows:

```
>SET /UIC=[libuic]
```

The [libuic] is determined by the system you are on. If you are on an RSX-11M-PLUS system, you can get the correct UIC with the command

```
>SET /LIBUIC
```


If you are on an RSX-11M system, the system UIC is [1,54].

Now build the compiler with the following command:

```
>TKB @[11,36]F7711M
```

Retain the edited command files in [11,36] for reference.

4.2.3 Building the OTS

To build the OTS, first set the UIC to [1,1]:

```
>SET /UIC=[1,1]
```

You can build the F77 OTS in either of two ways:

1. By merging the OTS concatenated object modules into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system, and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, invoke and use the LBR utility program as follows:

```
LBR>LB:SYSLIB.OLB=FP:[11,37]SHORT.OBJ/RP
LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=FP:[11,37]F4POTS.OBJ/RP
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
LBR>LB:SYSLIB.OLB=FP:[11,37]FCS11M.OBJ/RP
or
LBR>LB:SYSLIB.OLB=FP:[11,37]RMS11M.OBJ/RP
```

(See Section 2.3 for an explanation of the RMS-11 and the FCS-11 file systems.)

2. By generating a separate library using the OTS concatenated object modules. (If you build a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

That is, set the UIC to that of the new library and invoke the LBR utility program, as follows:

```
LBR>LB:F4POTS.OLB/CR:150.=FP:[11,37]SHORT.OBJ
LBR>LB:F4POTS.OLB/DG:$ERTXT
LBR>LB:F4POTS.OLB=FP:[11,37]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
LBR>LB:F4POTS.OLB=FP:[11,37]FCS11M.OBJ
or
LBR>LB:F4POTS.OLB=FP:[11,37]RMS11M.OBJ
```

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
>PIP LB:[1,1]=FP:[11,40]*.OBJ,*.ODL,*.MAC
```

Then copy the verification test program to the system disk as follows:

```
>PIP LB:[1,1]=FP:[11,40]*.FTN
```

4.2.4 Copying the Message File

At this point, set the UIC to [1,2] and copy the F77 compiler message file to the system disk, as follows:

```
>SET /UIC=[1,2]  
>PIP LB:[1,2]=FP:[1,2]F77COM.MSG
```

4.3 INVOKING THE COMPILER

You invoke the F77 compiler in either of two ways:

- If the compiler is installed as an MCR function (see Section 2.5.1), you invoke the compiler with the following command:

```
>F77
```

- If the compiler is not installed, you invoke the compiler as follows:

```
>RUN $F77
```


CHAPTER 5

RSTS/E INSTALLATION

This chapter describes the procedures for installing FORTRAN-77 on the RSTS/E operating system, using the RSTS/E BUILD procedure. If you are more familiar with RSX-11 installations and command language than you are with RSTS/E installations and command language, refer to Appendix A for an alternative installation of PDP-11 FORTRAN-77 on a RSTS/E system.

The basic installation procedure consists of the following:

1. Mounting (and in some cases copying to the system disk) the Autopatch Kit
2. Mounting the PDP-11 FORTRAN-77 distribution kit
3. Building the compiler and the Object Time System (OTS) with the BUILD program

The RSTS/E Autopatch Kit contains any patches that must be made to the PDP-11 FORTRAN-77 compiler and/or the Object Time System.

The Autopatch Kit and the distribution kit are contained on magnetic tapes or disk cartridges.

5.1 MOUNTING (AND COPYING) THE AUTOPATCH KIT

If your Autopatch Kit is on magnetic tape, follow the instructions in Section 5.1.1; if it is on disk, follow the instructions in Section 5.1.2.

5.1.1 Mounting and Copying a Magnetic-Tape Kit

Place the tape containing the Autopatch Kit, write-locked, on a tape drive; then place the tape on line. It is not necessary to specify the MOUNT command for magnetic tape.

Now use the PATCPY program to copy the Autopatch Kit to the system disk. Refer to the RSTS/E System Generation Manual for instructions on how to use the PATCPY program.

After the Autopatch Kit has been copied to the system disk, dismount the tape and refer to Section 5.2.

5.1.2 Mounting (and Copying) a Disk Kit

Place the disk cartridge containing the Autopatch Kit on an available disk drive, write-locked; then enter the command

```
MOUNT Dxn:PATCHE/RO
```

x

The value K, M, or L.

n

The number of the disk drive.

If a second disk drive is available, refer to Section 5.2; otherwise, follow the remaining instructions in this section.

If only one disk drive is available (the one on which the Autopatch Kit is mounted), you must now copy the Autopatch Kit to the system disk using the RSTS/E PATCPY program. Refer to the RSTS/E System Generation Manual for instructions on how to use the PATCPY program.

After the Autopatch Kit has been copied to the system disk, dismount the disk and refer to Section 5.2

5.2 MOUNTING THE F77 DISTRIBUTION KIT

To mount a magnetic tape distribution kit, place the tape, write-locked, on a tape drive; then place the tape on line. It is not necessary to specify the mount command.

To mount a disk distribution kit, place the disk cartridge on drive n, write-locked, and enter the command

```
MOUNT Dxn:F77/RO
```

x

The value K, M, or L.

n

The number of the disk drive.

5.3 BUILDING THE COMPILER AND THE OTS

You use the RSTS/E BUILD program to build (install) the compiler and the OTS.

First, log into a privileged account (for example, the one with PPN [1,0], but preferably not the one with PPN [1,2]); then type the following command and press **(RET)**:

```
RUN $BUILD
```

After you press **(RET)**, BUILD prints the following message:

```
BUILD V7.0-07 RSTS V7.0-07 Time Sharing
```


RSTS/E INSTALLATION

BUILD then prompts you for information by asking you a series of questions. These questions are given and discussed below:

SYSTEM BUILD <NO> ?

Press to accept the NO default.

SOURCE INPUT DEVICE <SY:> ?

Enter the mnemonic and unit number of the device on which you placed the FORTRAN-77 distribution medium, followed by a colon (:). For example, if you received your software on magnetic tape and mounted it on tape drive 0, you type MT0:.

LIBRARY OUTPUT DEVICE <SY:> ?

BUILD asks for the mnemonic name and unit number of the device containing the system library. Because the FORTRAN-77 compiler assumes the library is on SY: in account [1,2], it is more convenient to have the system library on that device. If your library is on SY:, press RETURN. If not, specify the name of the device that contains your system library.

TARGET SYSTEM DEVICE <SY0:> ?

Press to accept the SY0: default.

LIBRARY ACCOUNT <[1,2]> ?

The system library account contains programs that BUILD uses during the installation. BUILD must know the location of these programs. If the account number for your system library is [1,2], accept the default by pressing . If the library is in another account, type this account number in the form [p,pn].

CONTROL FILE IS ?

The control file, F77.CTL, contains the commands needed to install FORTRAN-77. To tell BUILD which file to use, type F77, then press .

BUILD then prints the following message on your terminal, telling you it is copying the F77.CTL file from the distribution medium to temporary file BLD01.TMP.

___ COPYING FILE dev: [1,2] F77.CTL TO BLD01.TMP ___

If you did NOT accept the SY: default to LIBRARY OUTPUT DEVICE, BUILD prints the question

LOCATE LOGICAL 'LB:' ON <SY: [1,1]> ?

It is recommended that you place LB: in an account other than [1,1]. Storing these files in [1,1] clutters the Master File Directory and slows access to user accounts and files. Therefore, type the disk device mnemonic, the unit number of the disk drive, a colon, and the account number you have chosen. The account you specify becomes the location of the logical name LB:.

FUNCTION (BUILD/PATCH, PATCH, BUILD) <BUILD/PATCH> ?

If you mounted the RSTS/E Autopatch Kit or have patches on disk, press to accept the <BUILD/PATCH> default. You can now install and

patch the software at the same time. BUILD installs FORTRAN-77 properly even if there are no patches to apply.

PATCH FILE INPUT LOCATION <SY:[200,200]> ?

Press if you transferred the patches from the distribution medium to account [200,200] on the public disk structure. If not, specify the device and account that contain the patches.

SAVE PATCHED SOURCES <NO> ?

Accept the NO default by pressing . The installation of FORTRAN-77 does not create patched sources. (BUILD may create patched sources when you patch other source code.)

You select features for the Object Time System by answering the following questions:

PUT THE F77OTS IN SYSLIB.OLB OR F4POTS.OLB <SYSLIB> ?

You can place the Object Time System in SYSLIB.OLB or F4POTS.OLB. Because the Task Builder assumes the OTS is in SYSLIB.OLB, placing it in SYSLIB.OLB is more convenient. With the OTS in F4POTS.OLB, however, you must explicitly specify this file when you task-build a program.

HAVE F77 USE FCS OR RMS BY DEFAULT <FCS> ?

The OTS cannot support both FCS and RMS at the same time; therefore, you must select either one or the other of these file systems. If you have need of indexed files, type RMS, then press . If you only need the features provided by FCS, accept the default. For more information on OTS options, see Chapters 2 and 3.

USE THE LONG OR SHORT ERROR TEXT MODULE BY DEFAULT <LONG> ?

If you choose the short error-text module, FORTRAN-77 creates complete error reports, but without the 1-line English description of the error condition. The PDP-11 FORTRAN-77 User's Guide contains a complete list of OTS error numbers and message text. If you want the long error-text module as the system default, press . Type SHORT if you prefer the short version.

ADDITIONAL CONTROL FILE IS <NONE> ?

Because there are no more control files to process, press . BUILD starts the installation. As BUILD processes the information from the distribution medium, it prints a log on your terminal, annotated by comment lines. These comments, prefixed by exclamation marks (!), give brief descriptions of each BUILD activity.

CHAPTER 6

INSTALLATION ON VAX/VMS UNDER AME

This chapter describes the procedures for installing PDP-11 FORTRAN-77 on a VAX/VMS (under AME) operating system.

The basic installation procedure for FORTRAN-77 consists of the following:

1. Building the F77 compiler task from an object module library
2. Building the F77 OTS library from object modules
3. Copying the compiler diagnostic message file from the distribution medium to the system disk

6.1 PREPARATION

To prepare for the installation of FORTRAN-77 on VAX/VMS under AME, proceed as follows:

1. Log into the system manager's privileged account while the system is stand-alone.
2. Be sure the logical name SYS\$DISK is assigned to the disk that contains the current version of VAX/VMS. This disk also contains the command procedure that initiates the new installation or update procedure. Note that SYS\$DISK need not be (and, if possible, should not be) SYS\$SYSTEM.
3. Set the following defaults in the order shown:

SET UIC [1,4]

SET DEF SYS\$SYSROOT:[SYSUPD]

6.2 BUILDING THE COMPILER

At the terminal, enter:

@VMSINSTAL 296 dev:

dev:

The system console device.

The VMSINSTAL procedure requires that you perform layered product installations according to the guidelines listed in Section 6.1. If you have not taken care to adhere to these guidelines, VMSINSTAL may type diagnostics on the console in the form:

%VMSINSTAL-W-DECNET, Your DECNET network is up and running.

If any diagnostic appears, the following question appears at your terminal:

* Do you want to continue anyway [NO] ?

Type (RET), and prepare for the installation according to the guidelines listed in Section 6.1. Then restart the installation procedure by starting at the beginning of this section.

The following message appears on your terminal when you invoke VMSINSTAL on a stand-alone system:

* Are you satisfied with the backup of your system disk [YES] ?

When you answer yes, the following message appears at the console:

Please mount the first volume of the set on dev:.

* Are you ready?

Place the first volume in the drive. Enter Y to proceed. A message appears at the console confirming that the distribution volume has been mounted. The following messages appear at your console:

The following products will be installed:

296 V5.0

Beginning installation of 296 V5.0 at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product saveset A...

For each additional piece of distribution medium, the installation procedure will print the following messages and query at the terminal:

%BACKUP-I-RESUME, resuming operation on volume 2

%BACKUP-I-READYREAD, mount volume 2 on dev: for reading

Press (RET) when ready:

Type (RET) to continue. Finally, when the last distribution medium is mounted, the following query appears at the terminal:

* Do you want to purge files replaced by this installation [YES]?

Type (RET) to have FORTRAN-77 files superseded by this installation purged. Enter NO if you do not want the superseded files purged.

The following message appears on the console before the compiler is task built:

The FORTRAN-77 compiler will be built now. This procedure will take approximately 15 minutes.

The procedure does not provide for on line changes to the compiler options command file.

The following warning diagnostics may appear on your terminal:

```
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=3
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=4
TKB -- *DIAG*-Task image file F77.EXE;n not contiguous
```

The last message indicates that the TKB facility tried unsuccessfully to allocate approximately 390 contiguous disk blocks on the system disk for the compiler image. Since the task image does not require contiguous disk blocks in order to run, the TKB allocates 390 noncontiguous disk blocks for the task.

6.3 BUILDING THE OTS

After the FORTRAN-77 compiler is built, the following messages about the library options, including error messages, appear on the terminal:

```
You must select either the FCS-11 or RMS-11 file management
system to be included in your FORTRAN-77 OTS. You must have
RMS-11 Version 1.8, or a later version, in order to use RMS.
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=3
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=4
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=3
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=4
Entry points deleted:
$ERTXT
```

The following error messages will appear at your terminal:

```
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=3
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=4
```

The following query will then appear at your terminal:

```
* Do you want to use the FCS-11 file system:
```

Enter YES if you want to include the FCS-11 file system in the FORTRAN-77 OTS. Enter NO if you want to include the RMS-11 file system in the FORTRAN-77 OTS.

Two more error messages will appear at your terminal:

```
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=3
%RSX-E-NOSUCHDEV, TKB-assigned device not found, SY0:, lun=4
```

6.4 INSTALLATION VERIFICATION

After the OTS is built, the following message appears at the terminal:

```
* Do you want to run the IVP after installation [YES]?
```

Press **(RET)** to perform the Installation Verification Procedure. The Installation Verification Procedure will compile, task-build, and run a sample FORTRAN-77 test program. When the program is compiled, the following error message should appear:

```
F77 -- ERROR 28-E Overflow while converting constant or constant
expression
[I = 71347] in module F77TST at line 6
F77 -- 1 Error F77TST.FTN
```

The following query will appear at the terminal asking what flavor of the file system you are using:

* Does the OTS use the FCS-11 file system? [Y/N]:

Enter Y if the OTS uses the FCS-11 file system. Enter N if the OTS uses the RMS-11 file system. After the task is built, the test program is run. The following program output will appear at the terminal:

```

INSTAL -- ERROR 73
Floating zero divide
at PC = xxxxxx
in "F77TST" at or after 6

```

```

PDP-11 FORTRAN-77 INSTALLATION SUCCESSFUL!
INSTAL -- STOP

```

FORTRAN-77 Installation Verification Passed

Successful installation of 296 V5.0 at hh:mm

6.5 CLEAN-UP

After installing FORTRAN-77, VMSINSTAL will type the following query at the terminal:

Enter the products to be installed from the next distribution volume set.
 * Products [EXIT]:

Press **(RET)** to exit the installation procedure. VMSINSTAL will type the following query at your terminal:

Please mount the console volume on dev:
 * Are you ready?

Mount the console volume in the console drive and type Y to continue. A message confirming the mounting of the console volume will appear on the terminal. VMSINSTAL terminates successfully by printing a final message:

VMSINSTAL procedure done at hh:mm

CHAPTER 7

SYSTEM BUILD VERIFICATION

The PDP-11 FORTRAN-77 distribution kit includes a simple test program to verify that the system is correctly installed and ready for use. The test program, F77TST.FTN, contains two intentional nonfatal errors, one occurring during compilation and one during execution. The two errors are documented in Section 7.2.

7.1 TEST PROGRAM

For VAX/VMS installation and the RSTS/E installation documented in Chapter 5, the test program is automatically copied from the distribution medium to the system disk. The test program is also automatically compiled, linked, and run. If you are performing either a VAX/VMS installation or the nonalternative RSTS/E installation, you do not need to read this section.

For RSX installations and the alternative RSTS/E installation (documented in Appendix A), the test program is copied from the distribution medium to the system disk when you build the OTS (see Sections 4.1.3 and 4.2.3). If you are following an RSX installation procedure or the alternative RSTS/E procedure, you need to compile, link, and execute F77TST.FTN as described in this section.

NOTE

If you are using a RSTS/E system, replace all occurrences of LB:[1,1] with LB: in the following commands.

MCR and RSTS/E Commands

The following command sequences show you how to compile, task-build, and execute the test program on RSX and RSTS/E systems.

Compile: F77 F77TST=F77TST/TR:ALL

Link: if using an OTS located in SYSLIB,

(if using FCS-11)

TKB F77TST/FP=F77TST

(if using RMS-11)

TKB F77TST/FP=F77TST, LB:[1,1]SYSLIB.OLB/LB, LB:[1,1]RMSLIB/LB

or, if a separate library is used,

(if using FCS-11)

TKB F77TST/FP=F77TST, LB:[1,1]F4POTS/LB

(if using RMS-11)

TKB F77TST/FP=F77TST, LB:[1,1]F4POTS/LB, LB:[1,1]RMSLIB/LB

Execute: RUN F77TST

7.2 TEST RESULTS

During compilation, an overflow error should occur on source line 6. During execution, a floating zero-divide error should be reported at line 8.

During execution on a non-FPP machine, the following run-time error is printed:

```
-- Exiting due to ERROR 7
Reserved instruction trap (SST4)
at PC=xxxxxx
```

7.3 VERIFICATION PROCEDURE FAILURE

The preceding verification procedure can fail if the FORTRAN-77 system is incorrectly installed. Possible failures include:

- The F77 compiler produces the message:

```
F77 -- FATAL 08 * COMPILER DYNAMIC MEMORY OVERFLOW
```

This message indicates that dynamic memory for the compiler was not correctly allocated by EXTTSK or INS/INC (see Section 3.1.4). Remove and reinstall the compiler with a correct INC value.

- If the compiler message file, LB:[1,2]F77COM.MSG (SY:[1,2]F77COM.MSG for RSTS/E), has not been transferred correctly, the compiler treats the compile-time overflow in F77TST as a fatal error. The compiler does not create object module F77TST.OBJ.
- For RSX-11M and RSX-11M-PLUS, the task exits with OTS error 2, "Task Initialization Failure," if the Task Builder /FP switch was omitted, or if the operating system does not support a floating-point processor.

CHAPTER 8

OTS RESIDENT LIBRARIES

This chapter describes how you can create and use OTS resident libraries. In particular, it discusses reasons for using (and limitations of) various types of resident libraries, including supervisor-mode libraries and cluster libraries.

A resident library has the following characteristics:

- It resides in memory. It must be installed before a task that references it can be installed or run.
- It can be shared by multiple tasks. However, it occupies virtual address space in each task which is linked to it.

The FORTRAN-77 OTS has the following general limitations:

- It does not contain position-independent code (PIC) and therefore cannot be built into a PIC resident library.
- It cannot be built into a supervisor-mode library. (However, a FORTRAN-77 OTS resident library can be linked to the FCS supervisor-mode library on RSX-11M-PLUS systems that support supervisor-mode libraries; see Section 8.1.1.)

OTS resident libraries are not supported for VAX/VMS systems.

For more information on resident libraries, refer to the Task Builder manual for your operating system.

8.1 TYPES OF RESIDENT LIBRARIES

There are two general types of resident libraries: noncluster and cluster. Within each of these two types, you can employ various schemes of organizing your library (or libraries), and you can choose either File Control Services (FCS) or Record Management Services (RMS). The sections that follow describe the various combinations possible and some of the considerations involved in choosing a resident library organization.

8.1.1 Noncluster Libraries

In a simple noncluster library, all of the library code takes up virtual address space in the task. That is, the size of the library (and hence the amount of address space it requires) consists of the total amount of space required by OTS routines and the file system (typically FCS) routines. A resident library of this type may be faster than an equivalent cluster library organization, but it also takes up much more space. You must use this organization, however, if

you have an RSX system that contains no support for memory mapping directives or for supervisor-mode libraries. (Section 8.2.2 contains a sample command file to build this kind of library.)

For RSX-11M-PLUS systems that support supervisor-mode libraries, you can link the OTS resident library with the supervisor-mode FCS library (FCSFSL). With this organization, only the OTS library takes up virtual address space in your task. When available, this configuration is recommended because it is slightly faster than the equivalent cluster organization. Note, however, that you cannot use this scheme to link with an RMS library. (Section 8.2.3 shows how to build this kind of library.)

8.1.2 Cluster Libraries

Cluster libraries are sets of two or more resident libraries that share the same portion of virtual memory. Conceptually, cluster libraries are like memory-resident overlays: The two or more cluster libraries form a single memory-resident overlay tree in your task's virtual address space.

When your task is linked with cluster libraries, only one of the two or more libraries is mapped by your task at one time. Therefore, the amount of virtual address space dedicated to libraries is equal to the largest of the cluster libraries, rather than to their total. When a call is made to a routine in a library other than the one currently mapped, the task automatically remaps to the new library. This process incurs some overhead; cluster libraries are slower than their noncluster or supervisor-mode counterparts.

Nevertheless, if your task uses RMS, a cluster library organization is recommended. Using this scheme, you can include all of RMS and a great deal of the FORTRAN-77 OTS in 8K words of your task's virtual address space. (Section 8.2.4 shows how to build this kind of library.)

If your task uses FCS, you should use a cluster library organization for these same reasons, unless your system supports supervisor-mode libraries. The supervisor-mode library (as described in the preceding section) gains for your task the same advantages as those of a cluster library, and is faster. (Section 8.2.5 shows how to build a resident library clustered with FCS.)

8.2 CREATING AN OTS RESIDENT LIBRARY

This section shows how to build the various kinds of OTS resident library described in Section 8.1. Section 8.2.1 outlines the general procedure that you follow to build any OTS resident library. Sections 8.2.2 through 8.2.5 provide specific instructions for the various libraries. These sections include Task Builder command files that will build appropriate libraries in most situations.

NOTE

The command files listed in this chapter do not link either of the two OTS error message modules, \$ERTXT or \$SHORT, into the resident library. You may include one of these modules when building your library if you wish to force long or short error message text to be used by programs that link to that library.

8.2.1 General Procedure

The following list contains the steps you follow to create an OTS resident library. The text that follows the list contains elaborations on some of the steps. Before you start performing these steps, you should have read Section 8.1 and decided what kind of resident library you need.

1. Edit the supplied file F4PRES.MAC, if necessary, to customize the contents of the OTS resident library to suit your needs. Further information on F4PRES.MAC appears following this list.
2. Assemble F4PRES.MAC, giving F4PRES.OBJ as the output file.
3. Use a text editor to create the appropriate Task Builder command file to build your resident library. (The command files in Sections 8.2.2 through 8.2.5 should work either as is or with slight modifications.) Make sure that the OTS object module library that you specify in the command file contains file system modules (either FCS or RMS) that match the file system you intend to use.
4. Invoke the Task Builder and pass it the command file you just created.
5. Inspect the map file resulting from the task build. If the resident library is too large or is not large enough, edit (or reedit) F4PRES.MAC and repeat the steps outlined above. (More information on recommended library size appears following this list.)
6. Purge any task, map, and STB files resulting from previous task builds.
7. Install the library in your system, following the instructions in the documentation for your particular operating system.

The file F4PRES.MAC (which is available on the FORTRAN-77 kit) contains global references to OTS entry points. The modules referenced in F4PRES.MAC will make up the OTS resident library. You can edit F4PRES.MAC to include modules that your tasks use frequently, or to exclude modules that are used infrequently. (Editing instructions are included in the file.) You can also use F4PRES.MAC as is; it will create an 8K word OTS resident library for clustering with RMS (RMSRES), or a 7K word library for clustering with FCS (FCSRES).

If you edit F4PRES.MAC, your goal should be to create an OTS resident library that appropriately balances the requirements of size and functionality. If your library is very large, the virtual address space available for your task may be unreasonably small. (A maximum library size of 8K words is recommended.) On the other hand, the Task Builder will place in your task the object code for any modules it references that are not in the OTS resident library; thus, it does not make sense to exclude commonly used modules from the library.

To make the best use of available virtual memory, the OTS resident library should be nearly equal to but slightly below a multiple of 4K words. Each time the size of the library exceeds a 4K multiple, an

additional APR is required; this has the effect of reducing the virtual address space available to the task by an entire 4K words. The following table illustrates this relationship:

Size in Words	Number of APRs	Size in Octal Bytes
4096	1	20000
8192	2	40000
12288	3	60000
16384	4	100000

In the case of an OTS resident library that will be clustered with a file system library, there is an additional consideration. Remember that, with clustered libraries, the virtual address space occupied by the libraries is equal to the size of the largest of the libraries. For example, if your OTS resident library occupies 4K words and the file system library occupies 8K words, the libraries occupy a full 8K words of virtual address space. In this situation, there is no advantage to limiting the size of your OTS resident library to 4K words; you might as well use the full 8K words and have a richer library.

The file system cluster libraries occupy space as follows:

File system	Library name	Number of APRs
FCS	FCSRES	1
RMS	RMSRES	2

When you create the command file that will build the OTS resident library, include a PAR option as follows:

PAR=pname:base:length

pname is the partition name; it must be the same as the name of the resident library. For RSX-11M systems, this partition must exist in the system; for RSX-11M-PLUS, the INSTALL command will place the library in another partition if the specified partition does not exist. The values you supply for base and length depend on the number of APRs that the resident library occupies, as follows:

Number of APRs	Base	Length
1	160000	20000
2	140000	40000
3	120000	60000
4	100000	100000

When you use these values, the OTS resident library will occupy the highest virtual addresses of your task.

8.2.2 Building a Noncluster Library with FCS Routines Included

The command files in this section build 8K-word OTS resident libraries that include FCS modules. Note that you must edit F4PRES.MAC in order to bring the size of the library down to 8K words.

RSX Systems

The following command file builds an 8K-word OTS resident library (F4PRES.TSK) that includes FCS modules referenced by the OTS:

```
F4PRES/-HD/LI/-PI,F4PRES/-SP/MA,F4PRES=F4PRES
LB:[1,1]F4POTS/LB
/
STACK=0
UNITS=0
PAR=F4PRES:140000:40000
//
```

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
LIBR=F4PRES:RO
```

RSTS/E Systems

The following command file builds an 8K-word OTS resident library (F4PRES.TSK) that includes FCS modules referenced by the OTS:

```
F4PRES/-HD/LI/-PI,F4PRES/-SP/MA,F4PRES=F4PRES
LB:F4POTS/LB
/
STACK=0
UNITS=0
PAR=F4PRES:140000:40000
//
```

Note that the FORTRAN-77 OTS object library (LB:F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
LIBR=F4PRES:RO
```

8.2.3 Building a Noncluster Library Linked to FCSFSL

The following command file builds an OTS resident library (F4PRES.TSK) for which FCS routines reside in a separate supervisor-mode library. As supplied, F4PRES.MAC builds a 7K-word library to link to FCSFSL. To make the best use of virtual address space, you can edit F4PRES.MAC (according to instructions that it contains) so that it takes up a full 8K words before you build this library. You can use this configuration only on RSX-11M-PLUS systems that support supervisor-mode libraries.

```
F4PRES/-HD/LI/-PI,F4PRES/-SP/MA,F4PRES=F4PRES
LB:[1,1]F4POTS/LB
/
STACK=0
UNITS=0
SUPLIB=FCSFSL:SV
PAR=F4PRES:140000:40000
//
```

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
LIBR=F4PRES:RO
```

8.2.4 Building a Library to Cluster with RMSRES

The command files in this section build 8K-word OTS resident libraries that cluster with RMSRES, the RMS file system resident library. You can use F4PRES.MAC as is to create this resident library.

NOTE

The ability to cluster a resident library with RMSRES is a new feature for RMS-11 V2.0. This configuration is not supported for installations still using RMS-11 V1.8 or earlier.

RSX Systems

The following command file builds an 8K-word OTS resident library (F4PCLS.TSK) that clusters with RMSRES:

```
F4PCLS/-HD/LI/-PI,F4PCLS/-SP/MA,F4PCLS=F4PRES
LB:[1,1]F4POTS/LB
LB:[1,1]RMSLIB/LB:RORMSC
/
STACK=0
UNITS=0
PAR=F4PCLS:140000:40000
GBLXCL=.SAVR1
//
```

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with RMS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
CLSTR=F4PCLS,RMSRES:RO
```

In addition, include the following RMS modules into the root of the task:

```
LB:[1,1]RMSLIB/LB:ROAUTL:ROIMPA:ROEXSY
```


RSTS/E Systems

The following command file builds an 8K-word OTS resident library (F4PCLS.TSK) that clusters with RMSRES:

```
F4PCLS/-HD/LI/-PI,F4PCLS/-SP/MA,F4PCLS=F4PRES
LB:F4POTS/LB
LB:RMSLIB/LB:RORMSC
/
STACK=0
UNITS=0
PAR=F4PCLS:140000:40000
GBLXCL=.SAVR1
//
```

Note that the FORTRAN-77 OTS object library (LB:F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with RMS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
CLSTR=F4PCLS,RMSRES:RO
```

In addition, include the following RMS modules into the root of the task:

```
LB:RMSLIB/LB:ROAUTL:ROIMPA:ROEXSY
```

8.2.5 Building a Library to Cluster with FCSRES

The following command file builds an 8K-word OTS resident library (F4PCLS.TSK) that clusters with FCSRES, the FCS file system resident library. As is, F4PRES.MAC builds a 7K-word library to cluster with FCSRES. By following instructions in F4PRES.MAC, you can edit it so that it uses the full 8K words.

OTS RESIDENT LIBRARIES

This command file is applicable to RSX systems only.

```
F4PCLS/-HD/LI/-PI,F4PCLS/-SP/MA,F4PCLS=F4PRES
LB:[1,1]F4POTS/LB
LB:[1,1]RMSLIB/LB:FCSVEC
/
STACK=0
UNITS=0
PAR=F4PCLS:140000:40000
GBLXCL=.FCSJT
GBLXCL=.CLOSE
GBLXCL=.CSI1
GBLXCL=.CSI2
GBLXCL=.DLFNB
GBLXCL=.FINIT
GBLXCL=.GET
GBLXCL=.GETSQ
GBLXCL=.GTDID
GBLXCL=.MRKDL
GBLXCL=.OPFNB
GBLXCL=.PARSE
GBLXCL=.POINT
GBLXCL=.POSRC
GBLXCL=.PRINT
GBLXCL=.PUT
GBLXCL=.PUTSQ
GBLXCL=.SAVR1
GBLXCL=.READ
GBLXCL=.WAIT
//
```

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
CLSTR=F4PCLS,FCSRES:RO
```


CHAPTER 9

RELEASE NOTES

9.1 FORTRAN-77 OTS WITH RMS-11 SUPPORT

The FORTRAN-77 RMS OTS that is installed on RSX-11M/M-PLUS and RSTS/E is compatible with RMS V2.0 that ships with RSX-11M V4.1, RSX-11M-PLUS V2.1, and RSTS/E V8.0. You must task-build these applications using RMS V2.0, since the RMS-OTS is not compatible with RMS V1.8.

The FORTRAN-77 RMS OTS that is installed on VAX/VMS is compatible with the component RMS-11 software, namely, RMS V1.8.

9.2 INSTALLING F77 ON RSTS/E

When using the automated RSTS/E installation procedure in Chapter 5, please note the following restriction before you answer the following prompt by the installation procedure:

PUT THE F77OTS IN SYSLIB.OLB OR F4POTS.OLB <SYSLIB> ?

You cannot install an OTS that supports the RMS file system in LB:SYSLIB.OLB if RMSLIB.OLB has already been installed in LB:SYSLIB.OLB.

9.3 F77 TYPING OF RESULT-GENERIC FUNCTIONS

Programmers should be aware of a new rule in the ANSI X3.9-1978 FORTRAN standard that affects the FORTRAN-77 compiler's behavior when typing result-generic functions. This rule, which is documented in the ANSI X3.9-1978 FORTRAN publication, page 8-5, states:

A type-statement that confirms the type of an intrinsic function whose name appears in the Specific Name column of Table 5 is not required, but is permitted. If a generic function name appears in a type statement, such an appearance is not sufficient by itself to remove the generic properties of that function.

That is to say, the following source program:

```
IMPLICIT INTEGER (A-Z)
INTEGER*4 NINT,I
REAL TEST
I=NINT(TEST)
END
```

does not cause the FORTRAN-77 compiler to automatically select the result type of the NINT function to be INTEGER *4.

The FORTRAN-77 rules governing the selection of the result type of integer-valued intrinsic functions are documented in Section 4.2.4 of the PDP-11 FORTRAN-77 User's Guide and are quoted here for convenience:

A number of intrinsic functions provided by FORTRAN-77 (for example, IFIX) produce integer results from real or double-precision arguments. These intrinsic functions are called "result-generic functions." Because the compiler operates in two different modes, INTEGER*2 mode and INTEGER*4 mode, the system provides two internal versions of each of these integer-producing functions: an INTEGER*2 version and an INTEGER*4 version. Selection of the proper version is made by the compiler mode setting, not, as it does for the other intrinsic functions, on the basis of the data type of arguments in the function reference.

Therefore, the result type of the NINT function is determined solely by the value of the I4 switch specified. When /NOI4 or /-I4 is specified for source compilation, the result type of the NINT function is INTEGER*2, not INTEGER*4.

9.4 COMPILER FAILS TO GENERATE WARNING DIAGNOSTIC

The compiler fails to generate ERROR-85 (name longer than 6 characters) for the following source program:

```
REAL TOLNUM,TOLSLT
READ (1)TOLNUMTOLSLT
END
```

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

9.5 ADJUSTABLE ARRAY USED IN INVALID CONTEXT

When encountering the following illegal use of an adjustable array, the FORTRAN compiler generates an expected compiler error 40 and an unexpected compiler FATAL*10 diagnostic:

```
SUBROUTINE BBBB
COMMON SIZE
INTEGER SIZE, ABC(SIZE)
END
```

As a temporary solution to this problem, you can modify the source program to avoid both the compiler error 40 and the FATAL*10 diagnostic by declaring the integer array ABE in the subroutine BBBB parameter list.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

9.6 INTEGER ARRAY REFERENCE USED AS INDEX IN BYTE ARRAY SUBSCRIPT

When the FORTRAN-77 compiler attempts to generate code for BYTE array subscripts and the subscript that is itself an INTEGER array reference, the compiler generates a FATAL*10 diagnostic.

RELEASE NOTES

The following source program causes the compiler to generate a FATAL*10 diagnostic:

```
      SUBROUTINE EXE(L)
      BYTE A1(240),A2(2100)
      INTEGER*2 T(100)
      DO 4300 L=1,N
      A1(T(L))=A2(T(L))
4300  CONTINUE
      END
```

As a workaround to this problem, compile the source program with the /NOOP switch.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

9.7 COMMON SUBEXPRESSIONS WITHIN CONDITIONAL STATEMENT

When the FORTRAN-77 compiler attempts to generate optimized code for the following source program, the compiler generates a FATAL*10 diagnostic:

```
      BYTE MINUS,ISGN1
      COMMON /DBUFF/ RBUFF(256)
      COMMON /OUTLN/LUN1
      DATA MINUS /'-'/
      IF(RBUFF(103).LT.0) ISGN1 = MINUS
      IF(RBUFF(104).LT.0) ISGN1 = MINUS
      IF(RBUFF(105).LT.0) ISGN1 = MINUS
      IF (RBUFF(101).EQ.3) WRITE (LUN1,*,ERR=950)
      1 ABS(RBUFF(103)),ABS(RBUFF(104))
      IF (RBUFF(101).EQ.4) WRITE (LUN1,*,ERR=950)
      1 ABS(RBUFF(103)),ABS(RBUFF(104)),
      1 ABS(RBUFF(105))
950  CONTINUE
      END
```

As a workaround to this problem, compile the source program with the /NOOP switch.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

9.8 RSTS/E INSTALLATION

When using the automated RSTS/E installation procedure in Chapter 5, put the system library on the system disk; that is, when the BUILD program issues the following prompt, respond by pressing the **(RET)** key:

```
LIBRARY OUTPUT DEVICE <SY:> ?
```

Otherwise, you may encounter a "Data error on device" message during the installation process.

APPENDIX A

ALTERNATIVE RSTS/E INSTALLATION

This appendix describes an alternative procedure for installing PDP-11 FORTRAN-77 on a RSTS/E operating system. If you are familiar with RSTS/E command language, you may wish to use the procedure described in Chapter 5, which uses the RSTS/E BUILD program; if you are not, but are familiar with RSX-11M/M-PLUS and command language, the following installation procedure may be more suitable.

The basic installation procedure for PDP-11 FORTRAN-77 consists of the following:

1. Building the F77 compiler task from an object-module library
2. Building a F77 OTS library from object modules
3. Copying the compiler diagnostic-message file from the distribution medium to the system disk

If your distribution kit is on magnetic tape, refer to Section A.1; if it is on disk cartridge, refer to Section A.2.

A.1 INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION

The following sections describe how to build PDP-11 FORTRAN-77 from a magnetic tape distribution kit.

A.1.1 Preparations

Before proceeding, check to see if the RT-11 and RSX run-time systems have been added. If they have not been added, add them as follows:

```
RUN SY:[1,2]UTILTY
#ADD RT11
#ADD RSX
```

Now place the distribution tape, write-locked, on a tape drive. Note that the device name for some tape units is MM: or MS:. If you are using such a device, substitute MM: or MS: for MT: in the commands that follow. If you are using 1600 BPI magnetic tape, you must mount the tape as follows:

```
MOUNT MT0:/DENSITY:1600
```

If your magnetic tape distribution is 800 BPI, you do not need to issue a specific mount command.

A.1.2 Building the Compiler

You build the compiler from object module library F77.OLB supplied on the distribution medium.

First, using a privileged account (for example, the account with PPN [1,0]; but preferably not the account with PPN [1,2]), copy the required files from the tape to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:*.*=MT0:[11,36]F77.OLB
*SY:*.*=MT0:[11,36]F77RST.*
```

You can now edit compiler task-build command file F77RST.CMD to select installation options, as described in Section 2.2 and Appendix B.

Now build the compiler as follows:

```
RUN SY:[1,2]TKB.TSK
TKB> @F77RST
```

Then copy the compiler to system library account [1,2], giving it protection code <104>:

```
RUN SY:[1,2]PIP.SAV
*SY:[1,2]F77.TSK<104>=F77.TSK
*SY:F77.TSK/DE (unless you are logged into [1,2])
```

Retain the edited command files for reference.

A.1.3 Building the OTS

To build the OTS, first copy the required OTS files from the tape to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:*.*=MT0:[11,37]*.OBJ
```

You can now build the F77 OTS in either of two ways:

1. By merging the OTS concatenated object modules into an existing library (for example, LB:SYSLIB.OLB). If you plan to use the RMS-11 file system, and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, invoke and use the LBR utility program as follows, making sure the module name and entry point tables can accommodate the OTS:

```
RUN SY:[1,2]LBR.TSK
LBR>LB:SYSLIB.OLB/CO:200.:2048.:512.=LB:SYSLIB.OLB
LBR>LB:SYSLIB.OLB=SHORT.OBJ/RP
LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=F4POTS.OBJ
```

Now use one of the following commands to incorporate RMS-11 or FCS-11 support. You must select only one of these file management systems:

```
LBR>LB:SYSLIB.OLB=RMS11M.OBJ
or
LBR>LB:SYSLIB.OLB=FCS11M.OBJ
```


ALTERNATIVE RSTS/E INSTALLATION

However, you may wish to build two separate OTS libraries, one using FCS-11 and one using RMS-11 (see Section 2.3).

2. By generating a separate library using the OTS concatenated object modules, as follows. (If you build a separate library, you must designate file LB:F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

```
RUN SY:[1,2]LBR.TSK
LBR>F4POTS.OLB/CR:150.=SHORT.OBJ
LBR>F4POTS.OLB/DG:$ERTXT
LBR>F4POTS.OLB=F4POTS.OBJ
```

Now use one of the following commands to incorporate RMS-11 or FCS-11 support, as follows:

```
LBR>F4POTS.OLB=RMS11M.OBJ
or
LBR>F4POTS.OLB=FCS11M.OBJ
```

You must select only one of these file management systems; however, you may wish to build two separate OTS libraries, one using FCS-11 and one using RMS-11 (see Section 2.3).

Copy the library to the library account LB:, giving it protection code <40>, as follows:

```
RUN SY:[1,2]PIP.SAV
*LB:F4POTS.OLB<40>=F4POTS.OLB
*F4POTS.OLB/DE
```

After completing one of the above, copy the optional OTS modules to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*LB:*.*=MT0:[11,40]*.OBJ,MT0:[11,40]*.ODL,MT0:[11,40]*.MAC/NO,
MT0:[11,40]*.FTN
```

A.1.4 Copying the Message File

At this point, copy the PDP-11 FORTRAN-77 compiler message file to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:[1,2]*.*=MT0:[1,2]F77COM.MSG
```

A.2 INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION

The following section explains how to build PDP-11 FORTRAN-77 from a disk cartridge distribution kit.

In the directions that follow, logical-unit assignment Dxn: to FP assigns the logical-device name FP: to the drive of the distribution disk. In the assignment, x can take the value K, M, or L, and n can take the value 0, 1, or any available drive. For example, use DK0 or DK1 for the RK05 distribution; use DM0 or DM1 for the RK06 or RK07 distribution; and use DL0 or DL1 for the RL01 or RL02 distribution.

ALTERNATIVE RSTS/E INSTALLATION

A.2.1 Preparations

Before proceeding, check to see if the RT-11 and RSX run-time systems have been added. If they have not been added, include the last two commands in the sequence below.

Now place the distribution disk cartridge in drive n (write-locked), mount the volume, and make the logical assignment shown below, as follows:

```
MOUNT Dxn:F77/RO

RUN SY:[1,2]UTILTY
#ADD LOGICAL Dxn: FP
#ADD RT11
#ADD RSX
```

A.2.2 Building the Compiler

You build the compiler from object module library F77.OLB supplied on the distribution medium. First, using a privileged account (for example, the account with UIC[1,1]; but preferably not the one with UIC[1,2]), copy the required files from the distribution disk to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:*.*=FP:[11,36]F77.OLB
*SY:*.*=FP:[11,36]F77RST.*
```

You can now edit compiler task-build command file F77RST.CMD to select installation options, as described in Section 2.2 and Appendix B.

Now build the compiler as follows:

```
RUN SY:[1,2]TKB.TSK
TKB> @F77RST
```

Then copy the compiler to system library account [1,2], giving it protection code <104>, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:[1,2]F77.TSK<104>=F77.TSK
*SY:F77.TSK/DE (unless you are logged into [1,2])
```

Retain the edited command files for reference.

A.2.3 Building the OTS

You can build the F77 OTS in either of two ways:

1. By merging the OTS concatenated object modules into an existing library (for example, LB:SYSLIB.OLB). If you plan to use the RMS-11 file system, and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, invoke and use the LBR utility program as follows, making sure the module name and entry point tables can accommodate the OTS:

```
RUN SY:[1,2]LBR.TSK
LBR>LB:SYSLIB.OLB/CO:200.:2048.:512.=LB:SYSLIB.OLB
LBR>LB:SYSLIB.OLB=FP:[11,37]SHORT.OBJ/RP
LBR>LB:SYSLIB.OLB/DG:$ERTXT
LBR>LB:SYSLIB.OLB=FP:[11,37]F4POTS.OBJ/RP
```

Now use one of the following commands to incorporate RMS-11 or FCS-11 support (see Section 2.3):

```
LBR>LB:SYSLIB.OLB=FP:[11,37]RMS11M.OBJ/RP
```

or

```
LBR>LB:SYSLIB.OLB=FP:[11,37]FCS11M.OBJ/RP
```

You must select only one of these file management systems; however, you may wish to build two separate OTS libraries, one using FCS-11 and one using RMS-11.

2. By generating a separate library using the OTS concatenated object modules, as follows. (If you build a separate library, you must designate the file LB:F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

```
RUN SY:[1,2]LBR.TSK
LBR>F4POTS.OLB/CR:150.=FP:[11,37]SHORT.OBJ
LBR>F4POTS.OLB/DG:$ERTXT
LBR>F4POTS.OLB=FP:[11,37]F4POTS.OBJ
```

Now use one of the following commands to incorporate RMS-11 or FCS-11 support. (See Section 2.3).

```
LBR>F4POTS.OLB=FP:[11,37]RMS11M.OBJ
```

or

```
LBR>F4POTS.OLB=FP:[11,37]FCS11M.OBJ
```

You must select only one of these file management systems; however, you may wish to build two separate OTS libraries, one using FCS-11 and one using RMS-11.

Copy the library to the library account LB:, giving it protection code <40>:

```
RUN SY:[1,2]PIP.SAV
*LB:F4POTS.OLB<40>=F4POTS.OLB
*F4POTS.OLB/DE
```

ALTERNATIVE RSTS/E INSTALLATION

After completing one of the above, copy the optional OTS modules to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*LB:*.*=FP:[11,40]*.OBJ,FP:[11,40]*.ODL,FP:[11,40]*.MAC
```

Copy the verification test program to the system disk, as follows:

```
*LB:*.*=FP:[11,40]*.FTN
```

A.2.4 Copying the Message File

At this point, copy the F77 compiler message files to the system disk, as follows:

```
RUN SY:[1,2]PIP.SAV
*SY:[1,2]*.*=FP:[1,2]F77COM.MSG
```

A.3 INVOKING THE COMPILER

You can invoke the F77 compiler in one of two ways:

- If installed as a CCL command (see Section 2.5.2), you invoke the compiler with the following:

```
F77
```

- If the compiler is not installed, you invoke it as follows:

```
RUN SY:[1,2]F77.TSK
```


APPENDIX B
COMPILER TASK-BUILD FILES

B.1 PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR RSX-11M/M-PLUS
(F7711M.CMD)

```

F77/CP/-FP,[11,36]F77/SP=[11,36]F7711M.ODL/MP
;
; PDP-11 FORTRAN-77 COMPILER TASK BUILD FILE
;
; PDP-11 FORTRAN-77 VERSION 5.0
;   RSX-11M VERSION 4.1
;   RSX-11M-PLUS VERSION 2.1
;
; SUMMARY OF SYSTEM PARAMETERS:
; REFERENCES PARTITION "GEN"
; 24k COMPILER TASK
; 512 WORD STACK
; 32 FRAMES IN EXPRESSION ANALYZER STACK
; 20 FRAMES IN DO/BLOCK IF STATEMENT STACK
; 45 COMMON BLOCKS, MAXIMUM
; 12 RESIDENT PAGES FOR WORKFILE SYSTEM
;
; OPTION INPUT
;
TASK      =...F77
; BUILD FOR PARTITION "GEN", MAPPED 11M SYSTEM
; PARTITION MUST BE AT LEAST 22K
;
PAR        =GEN
; SP STACK OF 512 WORDS
; STACK MUST NEVER BE LESS THAN 384 WORDS
;
STACK      =512
; F77 COMPILER LOGICAL UNIT ASSIGNMENTS
; 1  COMMAND INPUT
; 2  COMMAND OUTPUT
; 3  .OBJ OUTPUT
; 4  .LST OUTPUT
; 5  .FTN INPUT
;
; 6  COMPILER WORKFILE      (RANDOM ACCESS)
;   CAN BE REASSIGNED TO SWAPPING DISK IF AVAILABLE
;   DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
;   BUT THE WORKFILE DOES NOT REQUIRE A UFD ON THE VOLUME.
;
; 7  COMPILER TEMP FILES   (SEQUENTIAL ACCESS)
; 8  DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
;   BUT THE TEMP FILES DO NOT REQUIRE A UFD ON THE VOLUME.
;
; 9  COMPILER MESSAGE TEXT FILE
;
UNITS      =9

```

COMPILER TASK-BUILD FILES

```

ASG      =TI:1,TI:2
ASG      =SY0:6,SY0:7,SY0:8
ASG      =LB0:9
; RESIDENT MEMORY FOR WORKFILE VIRTUAL MEMORY SYSTEM
;
; UNDER RSX-11M/M-PLUS WORKFILE RESIDENT MEMORY IS DYNAMICALLY DETERMINED.
; IF THE OPERATING SYSTEM SUPPORTS DYNAMIC MEMORY ALLOCATION,
; THE SIZE OF THE COMPILER DYNAMIC STORAGE IS DETERMINED BY "EXTTSK".
; OTHERWISE, THE COMPILER WILL USE ALL MEMORY AVAILABLE IN THE PARTITION.
;
; INCREASING THE NUMBER OF RESIDENT WORKFILE PAGES WILL MAKE THE COMPILER
; RUN FASTER BY REDUCING PAGING I/O, BUT IT DOES NOT AFFECT THE SIZE
; OF THE MAXIMUM SOURCE PROGRAM WHICH CAN BE COMPILED.
;
EXTTSK    =3840
; F77 USES CONTROL SECTION "STACK1" FOR:
;   EXPRESSION ANALYZER STACK DURING PASS 1
;   NAMED COMMON BLOCK DEFINITIONS IN LATER PASSES
;
; AS DEFINED BELOW, STACK1 IS 312(10) WORDS, PROVIDING:
;   312/8   = 39 EXPRESSION ANALYZER STACK FRAMES
;   312/6   = 52 CONTROL SECTIONS
;           UP TO 7 CONTROL SECTIONS MAY BE USED FOR
;           COMPILER-GENERATED CODE AND DATA, LEAVING 45 COMMON BLOCKS.
;
EXTSCT    =STACK1:1160
; F77 USES CONTROL SECTION "DOSTK1" FOR:
;   DO STATEMENT NESTING STACK DURING PASS 1
;
; AS DEFINED BELOW, DOSTK1 IS 80(10) WORDS, PROVIDING:
;   80/4    = 20 NESTED DO/BLOCK IF STATEMENTS
;
EXTSCT    =DOSTK1:240
; DEFINE PRINTER WIDTH AND NUMBER OF SOURCE LINES PER LISTING PAGE
; F77 DEFAULT VALUES ARE:
;   55 SOURCE LINES PER PAGE (PLUS 3 LINES OF HEADING)
;   132 COLUMN LINE PRINTER
; NOTE:
;   55(10) = 67(8)
;   80(10) = 120(8)           132(10) = 204(8)
;
GBLPAT    =FORTRN:LPLINE:67
GBLPAT    =FORTRN:LPWDTH:204
; DEFINE DEFAULT OUTPUT FILE SUPERCEDE BEHAVIOR: A VALUE OF 0 (DEFAULT)
; INDICATES THAT THE COMPILER SHOULD NOT SUPERCEDE OUTPUT LISTING AND
; OBJECT FILES; A VALUE OF 1 ALLOWS SUPERCEDING.
;
GBLPAT    =COMAND:SUP00:0
; DEFINITION OF COMPILER SWITCH OPTION VALUES
;
; A COMPLETE DESCRIPTION OF THE EFFECTS OF THE COMPILER OPTION SWITCHES
; IS CONTAINED IN SECTION 1.2 OF THE PL. 11 FORTRAN-77 USER'S GUIDE.
;
; SWITCH    SWITCH    VALUE TO GBLPAT
; NAME      SETTING
; -----
; CK        /-CK      0
;           /CK        1      ARRAY SUBSCRIPT BOUNDS CHECKING
;
; CO        /CO:19.   23      NUMBER OF CONTINUATION LINES
;           /CO:N.     N
;
; DE        /-DE      0
;           /DE        1      INCLUDE DEBUG LINES
;

```


COMPILER TASK-BUILD FILES

```

; I4          /-I4          0      DEFAULT INTEGER2
;          /I4          1      DEFAULT INTEGER4
;
; LA          /-LA          0      REINITIALIZE SWITCHES
;          /LA          1
;
; LI          /LI:0          0
;          /LI:1          1      SOURCE
;          /LI:2          2      SOURCE, MAP
;          /LI:3          3      SOURCE, MAP, GENERATED CODE
;
; RO          /-RO          0      R/W CODE SECTIONS
;          /RO          1      R/O CODE SECTIONS
;
; SP          /-SP          0      NO SPOOLING
;          /SP          1      SPOOLING
;
; TR          /-TR          0
;          /TR:NONE          0
;          /TR:NAMES          1
;          /TR:BLOCKS          3
;          /TR:ALL          7
;          /TR          7
;
; WF          /WF:2          2      NUMBER OF TEMPORARY FILES
;          /WF:N          1,2,3
;
; WR          /-WR          0      NO OPTIONAL WARNINGS
;          /WR          1
;
; F77          /-F77          0      FORTRAN 66 INTERPRETATION
;          /F77          1      FORTRAN 77 INTERPRETATION
;
; ST          /-ST          0
;          /ST:NONE          0
;          /ST:SOURCE          1
;          /ST:SYNTAX          2
;          /ST          2
;          /ST:ALL          3
;
; DB          /-DB          0      NO DEBUG INFORMATION
;          /DB          1      PRODUCE DEBUG INFORMATION
;
; EX          /-EX          0      72 COLUMN PER SOURCE LINE
;          /EX          1      132 COLUMN PER SOURCE LINE
;
; OP          /-OP          0      NO CODE OPTIMIZATION
; OP          /OP          3      WITH CODE OPTIMIZATION
;
; THE FOLOWING "GBLPAT" DEFINITIONS EFFECT DEFAULTS OF:
;
; /-CK/CO:19./-DB/-DE/-EX/-I4/-LA/LI:2/OP/-RO/SP/TR:BLOCKS/WF:2/WR/F77/-ST
;
; DEFAULT VALUES FOR SWITCH "XX" ARE DEFINED
; BY A "GBLPAT" TO GLOBAL VARIABLE "XX000".
GBLPAT      =FORTRN:LA000:0
GBLPAT      =COMAND:CK000:0
GBLPAT      =COMAND:CO000:23
GBLPAT      =COMAND:DE000:0
GBLPAT      =COMAND:I4000:0
GBLPAT      =COMAND:LI000:2
GBLPAT      =COMAND:RO000:0
GBLPAT      =COMAND:SP000:1
GBLPAT      =COMAND:TR000:3
GBLPAT      =COMAND:WF000:2

```

COMPILER TASK-BUILD FILES

```

GBLPAT      =COMAND:WR000:1
GBLPAT      =COMAND:F7700:1
GBLPAT      =COMAND:ST000:0
GBLPAT      =COMAND:DB000:0
GBLPAT      =COMAND:EX000:0
GBLPAT      =COMAND:OP000:3
/

```

B.2 PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR RSTS/E (F77RST.CMD)

```

F77/CP/-FP, F77= F77RST.ODL/MP
;
; PDP-11 FORTRAN-77 COMPILER TASK BUILD FILE
;
; PDP-11 FORTRAN-77 VERSION 5.0
; RSTS/E VERSION 7.2
;
; SUMMARY OF SYSTEM PARAMETERS:
; REFERENCES PARTITION "GEN"
; 24K COMPILER TASK
; 512 WORD STACK
; 39 FRAMES IN EXPRESSION ANALYZER STACK
; 20 FRAMES IN DO/BLOCK IF STATEMENT STACK
; 45 COMMON BLOCKS, MAXIMUM
; 12 RESIDENT PAGES FOR WORKFILE SYSTEM
;
;
; OPTION INPUT
;
TASK      =...F77
; The RSX-11M Emulator must
; BUILD FOR PARTITION "GEN", MAPPED 11M SYSTEM
; PARTITION MUST BE AT LEAST 28K
;
PAR        =GEN
; SP STACK OF 512 WORDS
; STACK MUST NEVER BE LESS THAN 384 WORDS
;
STACK      =512
; F77 COMPILER LOGICAL UNIT ASSIGNMENTS
; 1  COMMAND INPUT
; 2  COMMAND OUTPUT
; 3  .OBJ OUTPUT
; 4  .LST OUTPUT
; 5  .FTN INPUT
;
; 6  COMPILER WORKFILE      (RANDOM ACCESS)
;     CAN BE REASSIGNED TO SWAPPING DISK IF AVAILABLE
;     DISK MUST BE MOUNTED AS WRITABLE VOLUME,
;     BUT THE WORKFILE DOES NOT REQUIRE A UFD ON THE VOLUME.
;
; 7  COMPILER TEMP FILES * (SEQUENTIAL ACCESS)
; 8  DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
;     BUT THE TEMP FILES DO NOT REQUIRE A UFD ON THE VOLUME.
;
; 9  COMPILER MESSAGE TEXT FILE
;
UNITS      =9
ASG        =TI:1,TI:2
ASG        =SY0:6,SY0:7,SY0:8
ASG        =SY:9
; RESIDENT MEMORY FOR WORKFILE VIRTUAL MEMORY SYSTEM
;

```


COMPILER TASK-BUILD FILES

```

;
; INCREASING THE NUMBER OF RESIDENT WORKFILE PAGES WILL MAKE THE COMPILER
; RUN FASTER BY REDUCING PAGING I/O, BUT IT DOES NOT AFFECT THE SIZE
; OF THE MAXIMUM SOURCE PROGRAM WHICH CAN BE COMPILED.
;
EXTTSK      =3840
; F77 USES CONTROL SECTION "STACK1" FOR:
;   EXPRESSION ANALYZER STACK DURING PASS 1
;   NAMED COMMON BLOCK DEFINITIONS IN LATER PASSES
;
; AS DEFINED BELOW, STACK1 IS 312(10) WORDS, PROVIDING:
;   312/8   = 39 EXPRESSION ANALYZER STACK FRAMES
;   312/6   = 52 CONTROL SECTIONS
;           UP TO 7 CONTROL SECTIONS MAY BE USED FOR
;           COMPILER-GENERATED CODE AND DATA, LEAVING 45 COMMON BLOCKS.
;
EXTSCT      =STACK1:1160
; F77 USES CONTROL SECTION "DOSTK1" FOR:
;   DO STATEMENT NESTING STACK DURING PASS 1
;
; AS DEFINED BELOW, DOSTK1 IS 80(10) WORDS, PROVIDING:
;   80/4    = 20 NESTED DO/BLOCK IF STATEMENTS
;
EXTSCT      =DOSTK1:240
; DEFINE PRINTER WIDTH AND NUMBER OF SOURCE LINES PER LISTING PAGE
; F77 DEFAULT VALUES ARE:
;   55 SOURCE LINES PER PAGE (PLUS 3 LINES OF HEADING)
;   132 COLUMN LINE PRINTER
; NOTE:
;   55(10) = 67(8)
;   80(10) = 120(8)           132(10) = 204(8)
;
GBLPAT      =FORTRN:LPLINE:67
GBLPAT      =FORTRN:LPWDTH:204
; DEFINE DEFAULT OUTPUT FILE SUPERCEDE BEHAVIOR: A VALUE OF 0 (DEFAULT)
; INDICATES THAT THE COMPILER SHOULD NOT SUPERCEDE OUTPUT LISTING AND
; OBJECT FILES; A VALUE OF 1 ALLOWS SUPERCEDING.
;
GBLPAT      =COMAND:SUP00:0
; DEFINITION OF COMPILER SWITCH OPTION VALUES
;
; A COMPLETE DESCRIPTION OF THE EFFECTS OF THE COMPILER OPTION SWITCHES
; IS CONTAINED IN SECTION 1.2 OF THE PDP-11 FORTRAN-77 USER'S GUIDE.
;
; SWITCH      SWITCH      VALUE TO GBLPAT
; NAME        SETTING
; -----
; CK          /-CK        0
;              /CK        1      ARRAY SUBSCRIPT BOUNDS CHECKING
;
; CO          /CO:19.     23      NUMBER OF CONTINUATION LINES
;              /CO:N.      N
;
; DE          /-DE        0
;              /DE        1      INCLUDE DEBUG LINES
;
; I4          /-I4        0      DEFAULT INTEGER2
;              /I4        1      DEFAULT INTEGER4
;
; LA          /-LA        0      REINITIALIZE SWITCHES
;              /LA        1
;
; LI          /LI:0       0
;              /LI:1       1      SOURCE
;              /LI:2       2      SOURCE, MAP

```

COMPILER TASK-BUILD FILES

```

;          /LI:3          3          SOURCE, MAP, GENERATED CODE
;
; RO          /-RO          0          R/W CODE SECTIONS
;          /RO          1          R/O CODE SECTIONS
;
; SP          /-SP          0          NO SPOOLING
;          /SP          1          SPOOLING
;
; TR          /-TR          0
;          /TR:NONE          0
;          /TR:NAMES          1
;          /TR:BLOCKS          3
;          /TR:ALL          7
;          /TR          7
;
; WF          /WF:2          2          NUMBER OF TEMPORARY FILES
;          /WF:N          1,2,3
;
; WR          /-WR          0          NO OPTIONAL WARNINGS
;          /WR          1
;
; F77          /-F77          0          FORTRAN 66 INTERPRETATION
;          /F77          1          FORTRAN 77 INTERPRETATION
;
; ST          /-ST          0
;          /ST:NONE          0
;          /ST:SOURCE          1
;          /ST:SYNTAX          2
;          /ST          2
;          /ST:ALL          3
;
; DB          /-DB          0          NO DEBUG INFORMATION
;          /DB          1          PRODUCE DEBUG INFORMATION
;
; EX          /-EX          0          72 COLUMN PER SOURCE LINE
;          /EX          1          132 COLUMN PER SOURCE LINE
;
; OP          /-OP          0          NO CODE OPTIMIZATION
;          /OP          3          WITH CODE OPTIMIZATION
;
;
; THE FOLOWING "GBLPAT" DEFINITIONS EFFECT DEFAULTS OF:
;
; /-CK/CO:19./-DB/-DE/-EX/-I4/-LA/LI:2/OP/-RO/SP/TR:BLOCKS/WF:2/WR/F77/-ST
;
; DEFAULT VALUES FOR SWITCH "XX" ARE DEFINED
; BY A "GBLPAT" TO GLOBAL VARIABLE "XX000".
GBLPAT      =FOTRN:LA000:0
GBLPAT      =COMAND:CK000:0
GBLPAT      =COMAND:CO000:23
GBLPAT      =COMAND:DE000:0
GBLPAT      =COMAND:I4000:0
GBLPAT      =COMAND:LI000:2
GBLPAT      =COMAND:RO000:0
GBLPAT      =COMAND:SP000:1
GBLPAT      =COMAND:TR000:3
GBLPAT      =COMAND:WF000:2
GBLPAT      =COMAND:WR000:1
GBLPAT      =COMAND:F7700:1
GBLPAT      =COMAND:ST000:0
GBLPAT      =COMAND:DB000:0
GBLPAT      =COMAND:EX000:0
GBLPAT      =COMAND:OP000:3
//

```


COMPILER TASK-BUILD FILES

B.3 PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR VAX/VMS UNDER AME

```

F77/CP/-FP=F77VAX.ODL/MP
;
; PDP-11 FORTRAN-77 COMPILER TASK BUILD FILE
;
; PDP-11 FORTRAN-77 VERSION 5.0, VAX VMS VERSION 3.0
;
; SUMMARY OF SYSTEM PARAMETERS:
; REFERENCES PARTITION "GEN"
; 22K COMPILER TASK
; 512 WORD STACK
; 39 FRAMES IN EXPRESSION ANALYZER STACK
; 20 FRAMES IN DO/BLOCK IF STATEMENT STACK
; 45 COMMON BLOCKS, MAXIMUM
; 6 RESIDENT PAGES FOR WORKFILE SYSTEM
;
; OPTION INPUT
;
TASK      =...F77
; BUILD FOR PARTITION "GEN", MAPPED 11M SYSTEM
; PARTITION MUST BE AT LEAST 20K
;
PAR       =GEN
; SP STACK OF 512 WORDS
; STACK MUST NEVER BE LESS THAN 384 WORDS
;
STACK     =512
; F77 COMPILER LOGICAL UNIT ASSIGNMENTS
; 1  COMMAND INPUT
; 2  COMMAND OUTPUT
; 3  .OBJ OUTPUT
; 4  .LST OUTPUT
; 5  .FTN INPUT
;
; 6  COMPILER WORKFILE      (RANDOM ACCESS)
; CAN BE REASSIGNED TO SWAPPING DISK IF AVAILABLE
; DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
; BUT THE WORKFILE DOES NOT REQUIRE A UFD ON THE VOLUME.
;
; 7  COMPILER TEMP FILES   (SEQUENTIAL ACCESS)
; 8  DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
; BUT THE TEMP FILES DO NOT REQUIRE A UFD ON THE VOLUME.
;
; 9  COMPILER MESSAGE TEXT FILE
;
UNITS     =9
ASG       =TI:1,TI:2
ASG       =WK0:6,WK0:7,WK0:8
ASG       =LB0:9
GBLPAT    =COMAND:MSG6WD:0
; RESIDENT MEMORY FOR WORKFILE VIRTUAL MEMORY SYSTEM
;
; UNDER RSX-11M WORKFILE RESIDENT MEMORY IS DYNAMICALLY DETERMINED.
; IF THE OPERATING SYSTEM SUPPORTS DYNAMIC MEMORY ALLOCATION,
; THE SIZE OF THE COMPILER DYNAMIC STORAGE IS DETERMINED BY "EXTTSK".
; OTHERWISE, THE COMPILER WILL USE ALL MEMORY AVAILABLE IN THE PARTITION.
;
; INCREASING THE NUMBER OF RESIDENT WORKFILE PAGES WILL MAKE THE COMPILER
; RUN FASTER BY REDUCING PAGING I/O, BUT IT DOES NOT AFFECT THE SIZE
; OF THE MAXIMUM SOURCE PROGRAM WHICH CAN BE COMPILED.
;
EXTTSK    =2048
; F77 USES CONTROL SECTION "STACK1" FOR:
; EXPRESSION ANALYZER STACK DURING PASS 1

```

COMPILER TASK-BUILD FILES

```

; NAMED COMMON BLOCK DEFINITIONS IN LATER PASSES
;
; AS DEFINED BELOW, STACK1 IS 312(10) WORDS, PROVIDING:
; 312/8 = 39 EXPRESSION ANALYZER STACK FRAMES
; 312/6 = 52 CONTROL SECTIONS
; UP TO 7 CONTROL SECTIONS MAY BE USED FOR
; COMPILER-GENERATED CODE AND DATA, LEAVING 45 COMMON BLOCKS.
;
EXTSCT =STACK1:1160
; F77 USES CONTROL SECTION "DOSTK1" FOR:
; DO STATEMENT NESTING STACK DURING PASS 1
;
; AS DEFINED BELOW, DOSTK1 IS 80(10) WORDS, PROVIDING:
; 80/4 = 20 NESTED DO/BLOCK IF STATEMENTS
;
EXTSCT =DOSTK1:240
; DEFINE PRINTER WIDTH AND NUMBER OF SOURCE LINES PER LISTING PAGE
; F77 DEFAULT VALUES ARE:
; 55 SOURCE LINES PER PAGE (PLUS 3 LINES OF HEADING)
; 132 COLUMN LINE PRINTER
; NOTE:
; 55(10) = 67(8)
; 80(10) = 120(8) 132(10) = 204(8)
;
GBLPAT =FORTRN:LPLINE:67
GBLPAT =FORTRN:LPWIDTH:204
; DEFINE DEFAULT OUTPUT FILE SUPERCEDE BEHAVIOR: A VALUE OF 0 (DEFAULT)
; INDICATES THAT THE COMPILER SHOULD NOT SUPERCEDE OUTPUT LISTING AND
; OBJECT FILES; A VALUE OF 1 ALLOWS SUPERCEDING.
;
GBLPAT =COMAND:SUP00:0
; DEFINE I- AND D-SPACE SUPPORT FOR RSX-11M-PLUS V2.1: A VALUE OF 1
; INDICATES THAT OBJECT MODULES OUTPUT BY THE COMPILER CAN BE USED
; FOR BUILDING I- AND D-SPACE TASKS; A VALUE OF 0 (DEFAULT) DOES NOT
; ALLOW THIS. THE VALUE MUST BE 0 IF FORTRAN-77 DEBUG IS USED.
;
GBLPAT =FORTRAN:DSPACE:0
; DEFINITION OF COMPILER SWITCH OPTION VALUES
;
; A COMPLETE DESCRIPTION OF THE EFFECTS OF THE COMPILER OPTION SWITCHES
; IS CONTAINED IN SECTION 1.2 OF THE PDP-11 FORTRAN-77 USER'S GUIDE.
;
; SWITCH SWITCH VALUE TO GBLPAT
; NAME SETTING -----
; -----
; CK /-CK 0
; /CK 1 ARRAY SUBSCRIPT BOUNDS CHECKING
;
; CO /CO:19. 23 NUMBER OF CONTINUATION LINES
; /CO:N. N (OCTAL VALUE)
;
; DE /-DE 0
; /DE 1 INCLUDE DEBUG LINES
;
; I4 /-I4 0 DEFAULT INTEGER2
; /I4 1 DEFAULT INTEGER4
;
; LA /-LA 0 REINITIALIZE SWITCHES
; /LA 1
;
; LI /LI:0 0
; /LI:1 1 SOURCE
; /LI:2 2 SOURCE, MAP
; /LI:3 3 SOURCE, MAP, GENERATED CODE
;

```


COMPILER TASK-BUILD FILES

```

; RO          /-RO          0      R/W CODE SECTIONS
;             /RO           1      R/O CODE SECTIONS
;
; SP          /-SP          0      NO SPOOLING
;             /SP           1      SPOOLING
;
; TR          /-TR          0
;             /TR:NONE      0
;             /TR:NAMES     1
;             /TR:BLOCKS    3
;             /TR:ALL       7
;             /TR           7
;
; WF          /WF:2         2      NUMBER OF TEMPORARY FILES
;             /WF:N         1,2,3
;
; WR          /-WR          0      NO OPTIONAL WARNINGS
;             /WR           1
;
; F77         /-F77         0      FORTRAN 66 INTERPRETATION
;             /F77          1      FORTRAN 77 INTERPRETATION
;
; ST          /-ST          0
;             /ST:NONE      0
;             /ST:SOURCE    1
;             /ST:SYNTAX    2
;             /ST           2
;             /ST:ALL       3
;
; DB          /-DB          0      NO DEBUG INFORMATION
;             /DB           1      PRODUCE DEBUG INFORMATION
;
; EX          /-EX          0      72 COLUMN PER SOURCE LINE
;             /EX           1      132 COLUMN PER SOURCE LINE
;
; OP          /-OP          0      NO CODE OPTIMIZATION
;             /OP           3      WITH CODE OPTIMIZATION
;
; THE FOLOWING "GBLPAT" DEFINITIONS EFFECT DEFAULTS OF:
;
; /-CK/CO:19./-DB/-DE/-EX/-I4/-LA/LI:2/OP/-RO/SP/TR:BLOCKS/WF:2/WR/F77/-ST
;
; DEFAULT VALUES FOR SWITCH "XX" ARE DEFINED
; BY A "GBLPAT" TO GLOBAL VARIABLE "XX000".
GBLPAT      =FORTRN:LA000:0
GBLPAT      =COMAND:CK000:0
GBLPAT      =COMAND:CO000:23
GBLPAT      =COMAND:DE000:0
GBLPAT      =COMAND:I4000:0
GBLPAT      =COMAND:LI000:2
GBLPAT      =COMAND:RO000:0
GBLPAT      =COMAND:SP000:0
GBLPAT      =COMAND:TR000:3
GBLPAT      =COMAND:WF000:2
GBLPAT      =COMAND:WR000:1
GBLPAT      =COMAND:F7700:1
GBLPAT      =COMAND:ST000:0
GBLPAT      =COMAND:DB000:0
GBLPAT      =COMAND:EX000:0
GBLPAT      =COMAND:OP000:3
/

```


READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

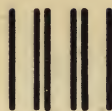
Organization _____

Street _____

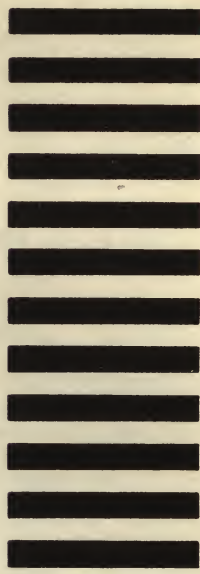
City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03061

Do Not Tear - Fold Here

Cut Along Dotted Line